

**TRABAJO FIN DE GRADO**

**Juego trivial en HTML5**

**Violeta Fraile  
Martín**

**Grado en ingeniería de sistemas  
audiovisuales**

*27/09/2015*

**Tutor: Yago Sáez**

# Trivial en HTML5

---

## Contenido

Resumen.....	6
1. Introducción y objetivos.....	11
1.1 Introducción al HTML.....	11
1.2 Historia de los videojuegos .....	13
1.3 Objetivos y motivación.....	14
2. Estado del arte .....	16
2.1 Planteamiento del problema .....	16
2.2 Análisis del estado del arte .....	17
2.2.1 HTML .....	17
2.2.2 CSS .....	18
2.2.3 JavaScript.....	20
2.2.4 PHP .....	23
2.2.5 MySQL .....	26
2.2.6 JQuery.....	28
2.2.7 Twitter Bootstrap .....	29
2.3 Páginas web educativas .....	31
2.3.1 Trivinet .....	31
2.3.2 Trivial Pursuit.....	32
2.3.3 Testeando.....	33
2.3.4 Triviados .....	35
2.3.5 Comparación de resultados .....	37
2.4 Marco regulador.....	39
3 Descripción del problema .....	41
4 Diseño y desarrollo de la solución.....	43
4.1 Implementación HTML.....	43
4.2 Implementación CSS.....	44

4.3	Implementación JavaScript .....	45
4.4	Implementación JQuery y MySQL .....	47
4.5	Tipos de preguntas .....	50
5	Organización del proyecto .....	51
5.1	Planificación del proyecto .....	51
5.2	Planificación inicial .....	51
5.3	Planificación final .....	53
5.3	Comparativa del trabajo estimado y realizado .....	55
5.4	Ciclo de vida del proyecto .....	56
6	Análisis económico .....	58
6.1	Herramientas hardware .....	58
6.2	Herramientas software .....	59
6.3	Gastos indirectos .....	59
6.4	Gastos de personal .....	59
6.5	Resumen de presupuesto .....	60
7	Análisis de la forma de venta de la aplicación .....	62
7.1	Modelo de pago .....	62
7.2	Modelo de suscripción .....	63
7.3	Modelo por publicidad .....	63
7.4	Modelo de compras dentro de la aplicación .....	63
7.5	Usuarios a los que va dirigida .....	64
7.6	Calcular el LTV (Lifetime Customer Value) .....	65
8	Evaluación de la aplicación .....	66
8.1	Método de evaluación mediante cuestionario .....	66
8.2	Método de evaluación mediante pruebas .....	67
9	Conclusiones .....	70
9.1	Problemas encontrados .....	70
9.2	Líneas futuras de trabajo .....	71
10.	Alternativas de diseño .....	72
10.1	Aplicaciones nativas .....	72
10.2	Aplicaciones web .....	73
10.3	Aplicaciones híbridas .....	74
Anexo I:	Manual de usuario .....	76

Anexo II: Summary .....	85
Anexo II: Introduction and objectives .....	89
Anexo III: Conclusions .....	93
Referencias.....	95

## Índice de figuras

Ilustración 1: Spacewar! .....	13
Ilustración 2: cronología css .....	18
Ilustración 3: tabla de comparación de las diferentes versiones de CSS .....	20
Ilustración 4: trivinet .....	32
Ilustración 5: Trivial pursuit.....	33
Ilustración 6: Testeando.....	34
Ilustración 7: testeando, juego.....	34
Ilustración 8: Triviados .....	37
Ilustración 9: Pantalla principal.....	76
Ilustración 10: Pantalla para "Elegir un tema" .....	77
Ilustración 11: Pantalla de juego.....	78
Ilustración 12: Pantalla de guardar puntuaciones .....	79
Ilustración 13: Pantalla de instrucciones .....	80
Ilustración 14: Pantalla de información .....	81
Ilustración 15: Pantalla de puntuaciones.....	82
Ilustración 16: Pantalla de encuesta .....	83
Ilustración 17: Pantalla de configuración.....	84
Ilustración 18: Pantalla principal.....	86
Picture 19: Spacewar! .....	91

## Resumen

Este proyecto tiene dos objetivos principales:

1. Comprender las nuevas tecnologías web como HTML5, JavaScript, CSS, jQuery, PHP y MySQL. El desarrollo de este proyecto es un caso de uso o una prueba de concepto de dichas tecnologías dentro del campo de los videojuegos educativos, así como un ejemplo práctico de todas las posibilidades que realmente ofrecen.
2. Realizar un juego didáctico y entretenido al que podrán jugar niños de prácticamente todas las edades.

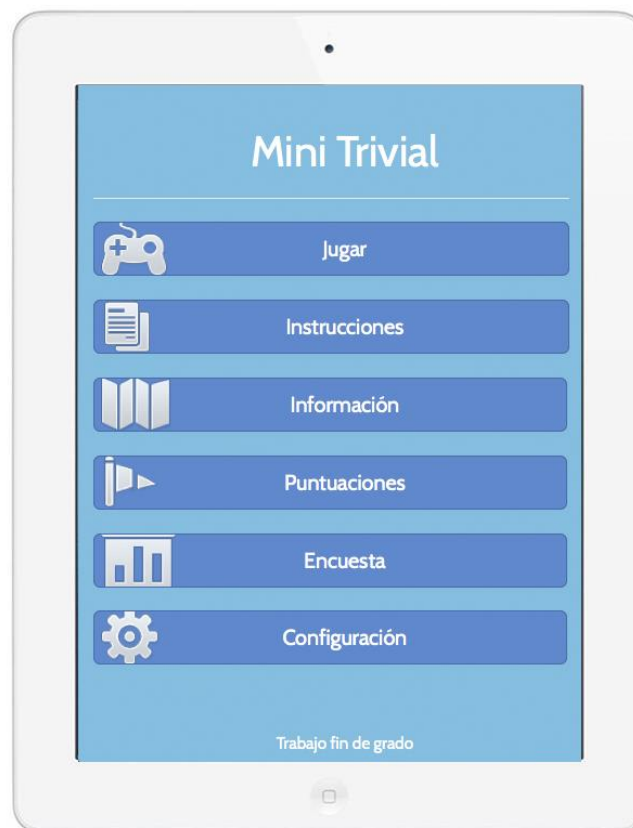
“Mini trivial” es un juego de preguntas y respuestas, tipo trivial, cuyas preguntas han sido obtenidas en su mayoría de libros de texto de 6º de primaria, por lo que los niños aprenderán a la vez que juegan.

En la actualidad, HTML se ha convertido en un lenguaje muy utilizado. Algunas de las ventajas que posee y que han sido en gran parte, motivo de su utilización en este proyecto son las siguientes:

- Es un lenguaje “opensource”, es decir, de código abierto y por tanto, independiente de terceros.
- Es semántico, con etiquetas que permiten clasificar y ordenar en distintos niveles y estructuras el contenido. Además, incorpora metadatos de manera más formal, favoreciendo el posicionamiento SEO (Search Engine Optimization) y la accesibilidad.
- El código es simple, lo que permite hacer páginas más ligeras que se cargan más rápidamente favoreciendo la usabilidad y la indexación en buscadores.
- Ofrece compatibilidad con los navegadores de dispositivos móviles.
- Incluye la etiqueta de dibujo canvas, que ofrece más efectos visuales.
- Ofrece soporte a codecs específicos.
- Posibilita la inserción de vídeos y audio de forma directa.
- Permite la geolocalización del usuario. Algo muy útil para el marketing móvil.
- Tiene la capacidad de ejecutar páginas sin estar conectado.

- Incorpora nuevas capacidades JavaScript que aumentan la capacidad de almacenamiento. Además, se permiten múltiples JavaScripts corriendo en paralelo en una misma página.
- Dispone de nuevas capacidades CSS3 como posibilidad de usar cualquier fuente o tipografía en HTML, columnas de texto, opacidad, transparencia, canales alpha, contraste, saturación, brillo, animaciones de transición y transformación, bordes redondeados, gradientes, sombras, etc.
- Permite realizar diseños adaptables a distintos dispositivos (web, tablets, móviles...).

En la pantalla principal, tal y como podemos ver en la imagen de abajo, tenemos los enlaces a todas las funcionalidades disponibles que se describirán a continuación.



Al pulsar sobre el botón de **“Jugar”** accedemos a una pantalla que nos permite seleccionar uno de los cuatro temas correspondientes a cada una de las asignaturas del curso académico de 6º de primaria.

Una vez seleccionada una de las materias, comenzará el juego en el cual nos podemos encontrar con 4 tipos de preguntas-respuestas:

1. Preguntas y respuestas escritas con texto. En este tipo de pregunta el usuario deberá elegir una de las 4 respuestas, pulsando sobre una de ellas.

Este es el primer tipo de pregunta/respuesta que se implementó y en el que se empleó más tiempo debido a la poca experiencia con algunos de los lenguajes utilizados para su implementación. El código fue modificado varias veces hasta conseguir el modo óptimo de mostrar los datos y pasar de una pregunta a la siguiente.

2. Pregunta escrita con texto y respuestas con fotografías. En este tipo de pregunta se deberá elegir una de las 4 fotografías que se proporcionará pulsando sobre el círculo (radio button) situado a la izquierda de cada foto.

Este tipo no fue tan sencillo de implementar como se pensó en un principio, en parte por el intervalo de tiempo transcurrido entre la implementación del tipo anterior y éste. Hubo además bastantes problemas a la hora de situar las imágenes para que vieran correctamente tanto en la Web como en los dispositivos móviles.

3. Pregunta y respuestas escritas con visualización de un vídeo. En este tipo de pregunta, lo primero que deberemos hacer es visualizar un vídeo, que contendrá la respuesta a la pregunta planteada más adelante. Tras pulsar el botón de “Continuar”, aparecerá la pregunta con las posibles respuestas y el usuario deberá seleccionar una de ellas.

Este tipo fue el más sencillo de todos gracias al gran avance de HTML5 en los elementos de vídeo. Se implementó una pantalla que mostrara primero el vídeo y tras pulsar en el botón de “Continuar”, pasase al primer tipo de preguntas/respuestas.

4. Puzle. En este tipo de pregunta aparecerá una imagen descolocada, que tendremos que ordenar, completando así el puzle.

Este tipo fue uno de los más difíciles de conseguir. No me resultó demasiado complicado lograr que funcionase para la parte web, conseguí que los trozos de las imágenes se movieran y se situaran en los recuadros habilitados. Sin embargo, dicho código no funcionaba para los dispositivos móviles puesto que el evento del clic del ratón es diferente al de la pantalla táctil. Conseguir capturar el evento de mover una imagen y sobre todo, de que dicha imagen se situara en el elemento habilitado, sin alterarse fue lo que más tiempo me llevó.



Todas las preguntas/respuestas descritas anteriormente tienen un temporizador. El usuario dispondrá del mismo periodo de tiempo para responder a los 3 primeros tipos de preguntas, pero un mayor periodo para responder al cuarto tipo (puzle). Esto es debido a que si este tipo durase lo mismo que los anteriores, la dificultad aumentaría demasiado para los niños.

Una vez contestada una pregunta, aparecerá un mensaje informativo en el que se mostrará si se ha respondido correctamente o no a dicha pregunta así como la que hubiera sido la respuesta correcta. En el tipo 2 (respuestas compuestas por imágenes) al responder de forma errónea, en el mensaje aparecerá la imagen que habría sido correcta.

En el caso de no haber elegido ninguna de las opciones y el tiempo se acabe, aparecerá también un mensaje informativo. Dicho mensaje mostrará el texto “Se ha quedado sin tiempo” así como la respuesta que habría sido la correcta.

Cabe mencionar que todas las imágenes que aparecen en las sucesivas preguntas y respuestas se pueden ver ampliadas gracias a la herramienta fancybox.


Al pulsar sobre el botón de **“Instrucciones”** situado en la página principal, se accederá a dicho apartado en el que los usuarios podrán consultar cualquier duda que tengan sobre la aplicación. En esta pantalla se mostrará mediante fotografías y de forma detallada las instrucciones para un correcto uso de la aplicación.

Por otro lado al pulsar sobre el botón de **“Información”** situado también en la página principal se accederá a un apartado en el que se mostrará un breve resumen acerca de quien ha sido el autor de la aplicación así como sus objetivos.

Además el juego consta de un sistema de puntuaciones. Al terminar cada ronda de 10 preguntas, los usuarios tienen la opción de guardar la puntuación obtenida introduciendo el nombre que deseen. Dichas puntuaciones se podrán visualizar más adelante en el apartado de **“Puntuaciones”**. De esta manera, se fomenta la competitividad de los usuarios y se incentiva a que sigan jugando.

Otro de los apartados de los que consta el juego es de un sistema de encuestas. Mediante la pantalla **“Encuesta”** los usuarios podrán puntuar una serie de preguntas que ayudarán a la mejora de la aplicación.

También se han implementado en la pantalla de **“Configuración”** un par de opciones que el usuario puede modificar a su gusto. La primera de ellas es la de “Elegir un idioma” en la cual el usuario tienen la posibilidad de elegir “Español” o “Inglés”. Una vez elegido uno de los idiomas, la web, incluida la parte de preguntas y respuestas



adoptará dicho idioma. La segunda de las opciones es la de “Elegir dificultad” que permite al usuario elegir una de las 3 dificultades modificando así la dificultad del juego.

Además, cabe mencionar que debido al gran crecimiento de las aplicaciones móviles, el juego ha sido desarrollado para un entorno multiplataforma, por lo que funciona correctamente tanto en la web como en dispositivos móviles.

# 1. Introducción y objetivos

## 1.1 Introducción al HTML

La Web ha cambiado mucho en los últimos años, hemos pasado de páginas sencillas, con texto, pocas imágenes y contenidos estáticos a páginas web más complejas, con contenidos dinámicos, diseño gráfico, multimedia e ingeniería. Es decir, las páginas Web actuales contienen servicios online y están basadas principalmente en la funcionalidad.

Brevemente, se puede definir una aplicación web como una herramienta en la que un usuario, a través de un ordenador, realiza peticiones a una aplicación remota por medio de Internet o de una intranet y recibe una respuesta, que es mostrada en el navegador del usuario.

HTML, cuyas siglas significan HyperText Markup Language, es un lenguaje sencillo y fácil de aprender, de marcado de hipertexto para la realización de páginas Web. A pesar de que actualmente hay más tecnologías que se pueden usar para programar las páginas Web como pueden ser VRML o Activex, los más utilizados son HTML y JavaScript. Esto se debe a que HTML es un estándar reconocido en todo el mundo, por lo que una página Web programada en dicho lenguaje se visualizará de forma muy parecida en cualquier navegador.

Un organismo llamado World Wide Web Consortium, también conocido como W3C, se encarga de conseguir dicha uniformidad, es decir, es el encargado de establecer dicho estándar. El W3C sacó la última versión llamada HTML5, en octubre del 2014, en la cual se establecieron nuevos elementos y atributos para cubrir las necesidades de las páginas Web actuales, dejando así obsoletos varios elementos de HTML 4.01.

Algunas de las mejoras más importantes introducidas en HTML5 son las siguientes etiquetas:

- `<video>` y `<audio>`: antes no había ninguna norma a la hora de mostrar videos o audios en una página Web y únicamente podían ser mostrados con algún plugin. Con esta mejora se crea un formato estándar para incrustar vídeos y audios en páginas Web.
- `<article>`, `<footer>`, `<header>`, `<nav>` y `<section>`: estas etiquetas surgieron de los usos que hacíamos de las etiquetas `<div>`, para organizar el contenido de la

Web y la necesidad de crear nuevas etiquetas que nos ayuden a organizar la estructura de una web en secciones.

- `<aside>`: con esta etiqueta podremos “rodear” el contenido que no es del contenido principal del que estamos hablando, es decir, se utilizará para elementos como pueden ser bloques publicitarios o eventos.
- `<time>`: esta etiqueta define una fecha o una hora en un formato establecido.
- `<input>`: se han añadido mejoras en esta etiqueta. Ahora se puede especificar el tipo de dato que esperamos que nos introduzcan en un campo de un formulario. Podemos definir los ingresos de texto como un entero, un email, una fecha, etc.
- `<dialog>`: esta etiqueta, como su propio nombre indica, define un cuadro de diálogo. De esta forma, podemos crear cuadros de diálogo emergentes con mayor facilidad.
- `<progress>`: con esta etiqueta podemos representar el progreso de una tarea que se esté ejecutando, suele utilizarse junto con JavaScript.
- `<canvas>`: es un contenedor para los gráficos, mediante esta herramienta podemos dibujar cualquier cosa y hacer animaciones complejas con la ayuda de JavaScript.

Información obtenida de [2].

## 1.2 Historia de los videojuegos

Los videojuegos tienen su origen a finales de la década de los 40, tras finalizar la Segunda Guerra Mundial, pero no es hasta la década de los 60 cuando aparecieron los primeros videojuegos modernos.

Actualmente hay diversas opiniones acerca de cuál fue el primer videojuego de la historia, repasemos la cronología:

- En 1946 se construyeron las primeras supercomputadoras.
- En 1947 Thomas T. Goldsmith y Estley Ray Mann patentaron un sistema electrónico que simulaba un lanzamiento de misiles contra un objetivo. Dicho proyecto no podría considerarse un videojuego, puesto que no había movimiento en pantalla.
- En 1952 Alexander S. Douglas desarrolló el Nought and crosses, también llamado **OXO**. Este juego era una versión computarizada de las tres en raya, en el cual se podía jugar contra la máquina.
- En 1958 William Higginbotham creó **Tennis for Two**, un simulador de tenis de mesa, en el que se representaba un campo de tenis y una red mediante líneas y una pelota.
- En 1961 Rusell crea el **Spacewar!**, juego en el que se representan dos naves que se enfrentan en el espacio exterior. En el juego, los jugadores podían controlar la dirección y la velocidad de las naves espaciales, además de poder luchar el uno contra el otro.

Actualmente, el Spacewar! es considerado por muchos el primer videojuego de la historia, puesto que el Tennis for Two solo usaba circuitería.

Desde que surgieron los primeros videojuegos han ido evolucionando constantemente, además hoy en día es uno de los sectores más rentables que hay.

Información obtenida de [3] y [4].



Ilustración 1: Spacewar!

## 1.3 Objetivos y motivación

Siempre he pensado que la motivación es un elemento clave en la educación, que no hay una mejor forma de aprender que jugando. Normalmente, los juegos contienen reglas y tienen un sistema de recompensa que los usuarios reciben al conseguir un objetivo. Las recompensas te van guiando hacia la meta final, de esta manera, un jugador puede ser consciente del nivel obtenido y de los fallos cometidos a pesar de no tener una calificación.

Debido a esto, decidí desarrollar un trivial, cuyas preguntas (en su mayoría) han sido obtenidas de libros de texto de 6º de primaria. Los motivos principales de elegir dicho juego son los siguientes:

- Al jugar, los niños podrán aprender las lecciones de sus libros de texto.
- Los niños, podrán comprobar inmediatamente después de contestar una pregunta si es correcta o no. Además, en caso de ser errónea, se les mostrará la respuesta correcta.
- Mediante el sistema de recompensas, en el trivial serían los “quesitos”, en el juego que he desarrollado son las puntuaciones, aumentará la motivación de los niños.
- Al ser preguntas que además de contener texto, contienen imágenes y vídeos, hace que el juego sea más ameno, convirtiéndose así en un modo bastante entretenido de aprender jugando.

Por otro lado, la realización de este proyecto tiene dos objetivos muy claros:

- La implementación de una herramienta que aporte novedades en el campo de la educación en internet.
- Ayudar a los niños a aprender de la misma forma que harían con libros de texto, pero de una forma más dinámica y divertida.
- Aprender los lenguajes de programación web para poder en un futuro dedicarme laboralmente a su desarrollo.

Finalmente, otra característica a tener en cuenta para motivar tanto a jugadores como estudiantes es desarrollar una relación positiva con los errores cometidos. En un videojuego aprendemos del fracaso porque este no es visto como un fin, sino como un paso necesario para llegar a un objetivo deseado y lograr así un aprendizaje. El trivial busca una relación positiva con el jugador, proporcionándole una retroalimentación

constante y manteniendo los episodios individuales de aprendizaje. En ambos casos, los errores se perciben como evidentes, por ejemplo, al contestar de forma errónea una pregunta evidencia el progreso de cada uno de forma individual, con el fin de generar una competitividad con uno mismo. Por este motivo es necesario ver claramente los errores cometidos. Por lo que en este proyecto, si un jugador contesta de forma errónea una pregunta, se le mostrará la opción correcta, para que pueda de esta manera aprender de los errores cometidos. Otra posible opción sería añadir explicaciones de las respuestas más detalladas o incluso enlaces a la respuesta correcta.

Los videojuegos, son eficaces para colocar al jugador dentro de una experiencia donde el interés se centra en el juego, ofrece la posibilidad de cambiar de identidad o se utilizan para animar a las personas a progresar a través de los contenidos y con el fin de motivar sus acciones. Estas técnicas pueden convertirse en una parte integral de los ambientes de aprendizaje formales porque pueden utilizarse en el aula con el fin de impulsar el compromiso de los estudiantes.

Dicho esto, me gustaría puntualizar que a pesar de que los videojuegos pueden servir de apoyo a la hora de animar a las personas a progresar a través de los contenidos, no deberían ser utilizados como único método educativo. Es decir, son una herramienta más para incentivar a los estudiantes a aprender.

## 2. Estado del arte

### 2.1 Planteamiento del problema

El objetivo del proyecto es la realización de un juego educativo, teniendo en cuenta cada una de sus etapas de desarrollo. El trivial es un juego que ha tenido diversas variaciones en los últimos años, la diferencia entre este proyecto y el resto es:

- Es una página web multiplataforma, podrá ser utilizado tanto para un entorno web como para uno móvil (Android/iOS).
- Está orientado especialmente a niños de 6º de primaria. La mayoría de las aplicaciones de tipo pregunta-respuesta que hay en el mercado están orientadas para niños con una edad superior.
- Tiene diferentes tipos de preguntas, no sólo las típicas preguntas con texto, como por ejemplo preguntas con imágenes o vídeos.

Al ser una página web orientada a niños, tiene un diseño sencillo y usable para su mejor utilización.



## 2.2 Análisis del estado del arte

A continuación explicaré cada una de las herramientas utilizadas en la realización del proyecto para una mejor comprensión del trabajo desarrollado.

### 2.2.1 HTML

Como ya se ha realizado una introducción a este lenguaje en la sección 1.1 no me extenderé demasiado en esta parte. Me centraré más en explicar cómo comenzar a utilizar esta tecnología.

Para comenzar la implementación de una página web con HTML deberemos tener dos herramientas:

- Un editor de texto. Puede servir cualquiera siempre que no formatee el texto. Los más utilizados son Microsoft Visual Studio Express, JetBrains WebStorm o Sublime Text.
- Un navegador. Lo utilizaremos para visualizar la página web escrita en HTML. Para poder interpretar correctamente las nuevas versiones de HTML, los desarrolladores de los navegadores web deberán incorporar dichos cambios. Normalmente los cambios son aplicados mediante parches de actualización automática (Firefox, Chrome). Pero otras veces, los cambios se aplican ofreciendo una nueva versión del navegador con todos los dichos cambios incorporados, en un sitio web de descarga oficial (Internet Explorer). Debido a esto, un navegador que no haya sido actualizado, no podrá interpretar correctamente una página Web escrita en una versión de HTML superior a la que puede interpretar. Por ese motivo, las páginas escritas en versiones anteriores de HTML deben ser actualizadas o reescritas. Sin embargo, esto no siempre se cumple, por lo que los navegadores mantienen la capacidad de interpretar páginas Web de versiones HTML antigua. Esta es la razón por la cual existen hoy en día diferencias entre los distintos navegadores al interpretar una página web.

Por lo tanto, utilizaremos el editor para ir escribiendo los documentos HTML y usaremos después el navegador para interpretar dicho documento e ir comprobando los cambios y añadidos que iremos efectuando.

Información obtenida de [8] y [9].

### 2.2.2 CSS

CSS es una hoja de estilo en cascada (Cascading Style Sheets), un lenguaje utilizado para definir y crear la presentación de un documento implementado en HTML o XML. Se denominan así porque la información de estilo puede situarse en varios lugares:

- En una hoja de estilo externa
- En una hoja de estilo incluida en la página Web
- Se pueden incluir los estilos dentro de las etiquetas de la página Web.



Ilustración 2: cronología css

Las hojas de estilo aparecieron en el año 1970, poco después del lenguaje de etiquetas SGML. Surgieron debido a la necesidad que apareció de aplicar de forma consistente diferentes estilos a los documentos electrónicos. Unos años más tarde, el organismo W3C (World Wide Web Consortium) propuso la creación de un lenguaje de hojas de estilo específico para el lenguaje HTML. De las propuestas que se presentaron, las que más se tuvieron en cuenta fueron la de CHSS (Cascading HTML Style Sheets) de Hakon Wium Lie y la de SSP (Stream-based Style Sheet Proposal) de Bert Bos. Entre 1994 y 1995 Lie y Bros se unieron y definieron un nuevo lenguaje con lo mejor de cada una de sus propuestas, dicho lenguaje lo llamaron CSS.

En 1995, el W3C decidió apostar por el desarrollo y estandarización de CSS y lo añadió a su grupo de trabajo de HTML. En diciembre de 1996, el W3C publicó su primera recomendación oficial, conocida como **“CSS nivel 1”** aunque unos años después, en 1999, se volvería a publicar como **“CSS nivel 1 revisada”** corrigiendo erratas y errores menores. A principios de 1997, W3C decidió separar los trabajos del grupo de HTML en tres secciones:

- Grupo de trabajo de HTML
- Grupo de trabajo de DOM
- Grupo de trabajo de CSS

En mayo de 1998, el grupo de trabajo de CSS publica su segunda recomendación oficial, conocida como **“CSS nivel 2”** y editada por Bert Bos y otros. Esta nueva

recomendación mantenía la compatibilidad con las hojas de estilo de nivel 1, incluía algunas características nuevas como el posicionamiento o los estilos de tablas y permitía la utilización de hojas de estilo adaptadas al medio de presentación (visual, oral, impreso, etc).

Hoy en día, los navegadores utilizan la versión de **CSS 2.1** que es una revisión de “CSS nivel 2”. La siguiente recomendación de CSS conocida como “**CSS nivel 3**” continúa en desarrollo desde 1998 y hasta el momento sólo se han publicado borradores.

La evolución de CSS se vio impulsada gracias al gran crecimiento de internet y del lenguaje HTML, así como a la falta de un estándar para la definición de los estilos. Dicha carencia de estándar dificultaba enormemente la creación de sitios web con diseño consistente en diferentes navegadores.

De esta forma, una página Web escrita en HTML o XML ya no contiene información acerca de la forma de representar la página, sino que esa información se encontrará ahora en la hoja de estilo (fichero .css). Las ventajas que se consiguen con la utilización de CSS son las siguientes:

- Uniformidad del diseño, gracias a que varias páginas HTML pueden utilizar la misma hoja de estilo.
- Facilidad en la modificación del estilo, gracias a que únicamente se tendrá que cambiar el diseño de una hoja de estilo y no de cada HTML.
- La amplia gama de herramientas de composición, más potentes que HTML.
- Puede ser utilizado con otros lenguajes de programación como por ejemplo JavaScript consiguiendo así efectos dinámicos en las páginas web.
- Podemos especificar hojas de estilo para distintos navegadores y para distintos dispositivos.
- Los usuarios con alguna discapacidad pueden definir su propia hoja de estilo y con la regla !important obligará al navegador a sustituir dicha hoja de estilo por la del autor de la página.

Por otro lado, la desventaja principal de CSS es la incompatibilidad de los distintos navegadores. Debido a esto, los diseñadores web deben conocer el soporte de CSS en cada uno de los navegadores más utilizados en el mercado.

Cada navegador, internamente está dividido en varios componentes. El componente del navegador más importante para un diseñador web es el motor, puesto que es el encargado de interpretar el código HTML y CSS para mostrar las páginas Web.

En la siguiente tabla analizaremos las compatibilidades de los navegadores más utilizados por los usuarios con las diferentes versiones de CSS:

Navegador	Motor	CSS 1	CSS 2.1	CSS 3
Google Chrome	WebKit	Completo desde la versión 85 del motor	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Internet Explorer	Trident	Completo desde la versión 7.0 del navegador	Completo	Todos los selectores, pseudo-clases y muchas propiedades a partir de la versión 10.0 del navegador
Firefox	Gecko	Completo desde la versión 1.0 del navegador	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Safari	WebKit	Completo desde la versión 85 del motor	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Opera	Presto	Completo desde la versión 1.0 del navegador	Completo	Todos los selectores, pseudo-clases y muchas propiedades

Ilustración 3: tabla de comparación de las diferentes versiones de CSS

Como podemos observar, los navegadores que mejor dan soporte a las versiones de CSS son Firefox, Chrome, Safari y Opera.

Información obtenida de [9], [10] y [11].

### 2.2.3 JavaScript

Abreviado comúnmente como “**JS**”, JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript, utilizado para crear páginas Web dinámicas, con acciones y animaciones. Además, al ser un lenguaje interpretado, no es necesario compilar los programas ejecutados, es decir, un programa escrito en JavaScript se puede probar directamente sin necesidad de procesos intermedios y sin tener que instalar ningún otro programa para su visualización.

Muchos confunden JavaScript con Java, pero ambos lenguajes son bastante diferentes, puesto que Java es un lenguaje más orientado a objetos.

Es necesario resaltar que hay dos tipos de JavaScript:

- **Navigator JavaScript.** Éste es el JavaScript propiamente dicho, se ejecuta en el cliente.
- **Live Wire JavaScript.** Éste es más reciente y se ejecuta en el servidor. Actualmente, se utiliza mucho para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX.

A principio de los años 90, empezaron a desarrollarse las primeras aplicaciones Web y a incluirse en ellas formularios más complejos, sin embargo, la velocidad de navegación de los usuarios era muy lenta (unos 28.8 kbps de máxima). Debido a este motivo, surgió la necesidad de un lenguaje de programación que se ejecutara en el navegador del usuario. Así si un usuario no rellenaba correctamente un formulario, gracias a JavaScript, en lugar de tener que esperar mucho tiempo la respuesta del servidor, se podría mostrar un mensaje de error sin la necesidad de hacer esa llamada.

Brendan Eich, un programador que trabajaba en Netscape, fue el desarrollador de JavaScript, que inicialmente recibió el nombre de Mocha. Un tiempo después, fue renombrado a LiveScript y en 1995, cuando Netscape agregó la compatibilidad de Java en su navegador Web Netscape Navigator, volvió a renombrarlo como JavaScript. Este último cambio de nombre provocó confusión, dando la impresión de que el lenguaje es una prolongación de Java y haciendo creer a muchos que la razón de este último cambio fue más que nada por marketing, puesto que Java era la palabra de moda de la época en el mundo informático y de Internet.

La primera versión de JavaScript fue un completo éxito y Netscape Navigator 3.0 ya incorporaba la siguiente versión del lenguaje, la versión 1.1. Al mismo tiempo, Microsoft lanzó JScript con su navegador Internet Explorer 3.0. JScript era una copia de JavaScript al que le cambiaron el nombre para evitar problemas relacionados con la marca. Sin embargo, las especificaciones de JScript eran incompatibles con la de ECMA en muchos aspectos.

En 1997 Netscape propuso que JavaScript fuera adoptado como estándar de la ECMA (European Computer Manufactures Association) enviándoles la especificación **JavaScript 1.1**.

Ese mismo año, fue aceptado como estándar ECMA con el nombre ECMAScript y poco después también como estándar ISO dando lugar al estándar ISO/IEC 16262.

ECMA ha publicado varios estándares relacionados con ECMAScript:



Ilustración 4: cronología ECMAScript

- **Primera edición.** Se publicó en 1997 del estándar ECMA-262.

- **Segunda edición.** Se publicó en 1998, esta edición se realizaron algunas modificaciones para adaptarlo al estándar ISO/IEC 16262.
- **Tercera edición.** Se publicó en 1999, esta es la versión que utilizan los navegadores actuales.
- **Cuarta edición.** Esta edición se abandonó, fue reducida en un proyecto cuyo nombre es ECMAScript Harmony.
- **Quinta edición.** Se presentó en el año 2009 con muchas nuevas características. Dos años después, en 2011, se lanzó ECMAScript 5.1. Esta nueva versión dejó de proporcionar algunas de las características anteriores y fue sobretodo una versión de mantenimiento para aclarar algunas especificaciones un poco ambiguas introducidas en la quinta edición.
- **Sexta edición.** Se publicó en Junio de 2015, aporta una mejora importante con respecto a la versión anterior. Añade características especiales, como algunos módulos, clases y mapas y además es compatible con las versiones anteriores.

De las muchas **características** de JavaScript, hay algunas de especial relevancia. A continuación se comentarán algunas de ellas puesto que hacen de este lenguaje algo singular para entender y comprender.

- JavaScript es un lenguaje orientado a objetos. Muchos desarrolladores se llevan cierta sorpresa al usarlo por primera vez, ya que en un principio no parece serlo.
- Algo muy importante y que puede parecer un poco extraño considerando el apartado anterior, es que en JavaScript no existen las clases. Para un desarrollador convencional esto es un tanto extraño y difícil de entender, sobre todo si se viene de desarrollos de los lenguajes más usados hasta el momento, algo que te obliga a desaprender para entender con mayor soltura.

Además, un objeto puede crearse y editarse en tiempo de ejecución, algo que con un desarrollo de clases no existe. A parte, en JavaScript es preferible la composición de objetos antes que la herencia de clases. Esto significa que un objeto es mejor crearlo con piezas de código que interesen de lo que se ha desarrollado, antes que hacer una aproximación más robusta de herencia padre-hijo.

- JavaScript es un lenguaje interpretado sin un tiempo de compilación, haciendo posible desplegar código o programas con errores. Por ello, muchas veces hace que el desarrollo sea un tanto difícil puesto que no dispone de un control mayor en su ejecución.

En sus inicios, para comprobar que el código que se desarrollaba era correcto no había otra forma que ver los resultados sobre el navegador. Actualmente

esta faceta ha sido mejorada y con los nuevos estándares se hace más liviano poder ejecutar código en los navegadores.

- Las funciones en JavaScript adquieren una especial relevancia. Pueden ser tanto propiedades de objetos, (porque en la asignación de clave-valor, el valor puede ser una función), como también pueden ser objetos propios, (también llamadas ‘funciones constructor’).

Como se pueden considerar como objetos, las funciones pueden crearse dinámicamente en tiempo de ejecución, pueden asignarse a variables, tienen sus propias propiedades y métodos e incluso se pueden pasar como referencia en los argumentos. Esto último es lo que permite una de las características más valoradas de JavaScript y que se comenta a continuación.

Como las funciones son objetos, estas pueden pasarse por argumento a otras funciones. Las funciones que reciben por argumento una referencia a otra función, pueden ejecutarlas, llamando a esto ‘callback’.

Es algo muy sencillo de utilizar, pero en JavaScript es muy poderoso a la hora de desarrollar librerías. Permite que el núcleo de las aplicaciones sea más liviano y sencillo de realizar, proveyendo de lo necesario a la función de retorno y ésta se encargará de lo necesario para hacer su uso.

- Con la idea del anterior punto, JavaScript es un lenguaje orientado al evento, haciendo que sea una de las características más importantes y usadas. La programación en los navegadores está orientado a los eventos asíncronos, haciendo que su uso sea muy extenso a la hora de realizar cualquier acción del usuario. Llamamos a un evento asíncrono a aquel que puede enviar datos en cualquier momento, sin previo aviso.

Información obtenida de [12].

## 2.2.4 PHP

PHP fue diseñado en un principio en Perl, con base en la escritura de un grupo de CGI binarios escritos en el lenguaje C por el programador danés Rasmus Lerdorf. Surgió cuando Rasmus se propuso escribir un pequeño CGI en Perl para que le ayudara a contabilizar las visitas que le llegaban a su currículum vitae, guardando ese valor en una base de datos MySQL. Posteriormente, Rasmus decidió reescribir su script en C puesto que el servidor web en el que se alojaba tenía algún problema de rendimiento. De esta forma, consiguió no tener que cargar un nuevo proceso de Perl cada vez que alguien visitaba su web.

Además el script de Rasmus llamó la atención de otras personas alojadas en su mismo servidor y le pidieron poder utilizarlo. Así, surge cada vez un mayor número de funcionalidades, que Rasmus decide unificar en una única librería C.

Lo siguiente que hace Rasmus es escribir un analizador sintáctico, cuya funcionalidad en un principio es buscar etiquetas en el código HTML, para que las reemplace con una de las salidas correspondientes de su pequeña librería. Más tarde, dicho analizador sintáctico irá aumentando en complejidad al añadir etiquetas condicionales, etiquetas de bucles y funciones. Rasmus llamó a ese paquete de software “Personal Home Page Tools”, más conocido como “PHP Tools”.

De forma paralela, Rasmus empezó a desarrollar otro CGI para analizar consultas SQL y facilitar así la creación de formularios y sus correspondientes tablas. Dicha herramienta recibió el nombre de FI (Form Interpreter) y al conjunto de ambos paquetes, se le conoció como PHP/FI.

En 1995 Rasmus anunció en las noticias la primera versión oficial de PHP bajo la licencia de GPL. Dicha versión únicamente corría en servidores UNIX y carecía de algunas características básicas, como la del bucle for. Después de esto, el número de usuarios empezó a crecer y algunos programadores empezaron a ayudar en el proyecto. El paquete se renombró a Personal Home Page Construction Kit al combinar PHP y FI y reescribirse el analizador sintáctico. Así, en 1997 se publica **PHP/FI 2.0** contando con unos 50.000 usuarios.

En 1998, apareció la **versión 3.0** con la que el lenguaje dio un salto muy grande en cuanto a rendimiento, funcionalidad y popularidad, llegando a alcanzar el millón de usuarios. Esto fue posible gracias a Zeev Suraski y Andi Gutmans, dos israelíes que encontraron PHP/FI 2.0 no del todo eficiente para lo que intentaban conseguir: una aplicación de comercio electrónico. Viendo el problema que se les presentaba, decidieron crear una nueva versión del analizador.

Zeev y Andi reescribieron el analizador sintáctico de PHP y ayudaron en la reescritura de buena parte del resto de su código. Además, atrajeron la atención de un gran número de programadores al diseñar una nueva API que permitía crear extensiones para PHP.

Otras novedades de esta versión, lanzada para Windows 95, Windows NT, Macintosh y varias versiones de UNIX, fue la introducción de una muy básica orientación a objetos, el soporte para los sistemas de bases de datos más populares, como MySQL y PostgreSQL, y la creación de un módulo nativo para Apache. Fue también en esta versión en la que el nombre del lenguaje perdió el sufijo /FI, quedando sólo en PHP, y en la que el acrónimo cambió su significado por el que conocemos hoy en día, ‘PHP Hypertext Pre-processor’. Esto ocurrió después de una votación en la que participaron



los desarrolladores, y tras un breve periodo de tiempo en que se utilizó como nombre 'Professional Home Pages'.

El 22 de Mayo de 2000 se lanza **PHP 4**, basado en el Zend Engine 1.0, una máquina virtual de código abierto, bajo licencia PHP, desarrollada también por Zeev y Andi. Desde esta versión, el código fuente se compila primero a un código máquina intermedio llamado bytecode, y el Zend Engine se encarga después de ejecutar este código, al estilo Java. Esto mejoró considerablemente la velocidad de ejecución respecto de la versión anterior.

Entre las novedades introducidas en PHP 4.0 también encontramos:

- Mejoras de rendimiento en la API de extensiones
- Una capa de abstracción que permitía a PHP ejecutarse en la mayoría de los servidores web más populares
- Sesiones HTTP
- Control de buffer de salida
- El tipo booleano
- Mejoras en la orientación a objetos y varias construcciones nuevas.

Además se pasó de usar una licencia dual GPL – PHP License, a utilizar exclusivamente la licencia PHP, porque Rasmus y el resto de desarrolladores no veían sentido a utilizar una licencia dual. Dicho cambio se realizó al tener en cuenta que los usuarios solían siempre utilizar la menos restrictiva de las dos. Sin embargo, este hecho causó un pequeño revuelo en la comunidad open source.

Las revisiones menores de PHP 4 también introdujeron mejoras interesantes, como la inclusión de las variables súper globales en **PHP 4.1.0**, la desactivación por defecto de la opción `register_globals` en **PHP 4.2.0**, o la posibilidad de ejecutar los scripts desde la línea de comandos y la capa de acceso a ficheros y recursos de red a través de flujos o streams en **PHP 4.3.0**.

En 2004, contando PHP con una base de usuarios de 15 millones de dominios, se lanza **PHP 5**, basado en el Zend Engine 2.0, versión que haría que PHP entrara en su madurez, y lo convertiría en un lenguaje muy agradable de utilizar. El esfuerzo principal a la hora de desarrollar esta versión se centró en una muy necesaria mejora en el soporte de la orientación a objetos, muy pobre y discutida hasta ese momento, aunque también se añadieron otras características igual de importantes, como la capa de abstracción de acceso a bases de datos PDO (PHP Data Objects u Objetos de Datos PHP), SQLite por defecto, manejo de excepciones con bloques try-catch, iteradores, mejoras en el manejo de XML gracias a la extensión SimpleXML, mejoras en la velocidad y el rendimiento, entre otras mejoras.

En el 2009 se lanzó la **versión 5.3**, que es la versión actual, introduciendo características muy interesantes, inicialmente pensadas para **PHP 6**, como los espacios de nombres, el late static binding, closures o funciones anónimas.

La futura versión PHP 6 viene desarrollándose desde el 2005, pero ha sufrido grandes retrasos debido a lo ambicioso del proyecto y los problemas con el soporte completo de Unicode. Tanto es así, que el desarrollo de PHP 6 se movió a una rama en Marzo de 2010, y el desarrollo continúa en el trunk, en un código basado en la versión 5.3.

No obstante, PHP cuenta con millones de usuarios en todo el mundo y se utiliza en un 75% de los servidores. Además, ha alcanzado el puesto de cuarto lenguaje más utilizado del mercado según TIOBE; con novedades tan útiles en un futuro próximo como los traits y con frameworks que nada tienen que envidiar a Django o Ruby on Rails.

Información obtenida de [13].

### 2.2.5 MySQL

MySQL es uno de los sistemas de gestión de bases de datos más utilizados del mundo, es relacional, multiusuario, multihilo y de código abierto. Su origen se remonta a la década de los ochenta, cuando Michael Widenius, un joven programador, realizaba complejas aplicaciones en lenguaje BASIC. Widenius estaba buscando un sistema de almacenamiento de archivos, pero al no encontrarlo, decidió crear el suyo propio. MySQL fue creado en 1995 por Michael Widenius, David Axmark y Allan Larsson, desarrolladores de una compañía sueca llamada MySQL AB. Su objetivo principal fue ofrecer opciones eficientes y fiables de gestión de datos para todo tipo de usuario. En el año 2000 se lanzaron más de seis versiones alfa y beta de la plataforma, la mayoría compatibles con las principales plataformas.

Fue a partir del año 2000 cuando MySQL empezó a ser de código abierto y a seguir los términos de GPL. Gracias a esto, sus ingresos disminuyeron significativamente, sin embargo, con el tiempo se recuperó y otros desarrolladores contribuyeron con el proyecto.

En el año 2001 MySQL había logrado bastante popularidad y tenía ya 2 millones de instalaciones activas. Por otro lado, en junio de ese mismo año, tuvo que enfrentarse a su primera demanda importante por violación de los contratos de terceros y competencia desleal, entre otras cosas, de parte de UnSphere en la Corte de Distrito de EE.UU. en Boston.

En 2002, la compañía se amplió y abrió una sede en EE.UU. y otra en Suecia. Ese mismo año, su número de miembros aumentó hasta sobrepasar los 3 millones de usuarios, con ingresos por valor de 6 millones y medio de dólares. Además, la empresa MySQL AB demandó a NuSphere por derechos de autor y violación de marca registrada. Esta demanda terminó con un acuerdo, el 27 de febrero, entre ambas compañías en la audiencia preliminar.

La empresa continuó ganando popularidad, por lo que a finales del 2003 podía presumir de tener un ingreso total de 12 millones de dólares, con 4 millones de instalaciones activas. Un año más tarde, la empresa decidió centrarse más en los ingresos del usuario final en lugar de en obtener una cuota de licencia por instalación. La estrategia que eligieron les salió rentable y al finalizar el año ya contaban con un ingreso neto de 20 millones de dólares.

En el año 2005, la empresa que gestionaba el almacenamiento backed Innobase de MySQL fue comprada por Oracle. Dicho motor de almacenamiento de MySQL permite la implementación de funciones importantes, como las transacciones y las claves foráneas. Ese mismo año, MySQL Network desarrolló en las líneas de RedHat, dando lugar a MySQL 5 y ampliando considerablemente el conjunto de características disponibles para los usuarios de la empresa. Dos años más tarde, el contrato entre MySQL e Innobase se revocó.

En 2006, Oracle compró también Sleepycat, la empresa que se encarga de la gestión del motor de almacenamiento transaccional Berkeley DB de MySQL. No obstante, Berkeley no fue muy utilizado y no se incluyó en las versiones de MySQL, por lo que dicha adquisición no fue muy importante. Mientras, la popularidad de la empresa seguía creciendo, alcanzando los 8 millones de instalaciones activas en el año 2006. Ese mismo año, MySQL tenía ya unos 320 empleados en 25 países. Dichos trabajadores contaban con una característica distintiva, el 70% de los trabajadores realizaban su trabajo desde casa gracias a la naturaleza de código abierto de la plataforma. Los ingresos de la compañía siguieron creciendo, consiguiendo un total de unos 50 millones de dólares a finales de 2006 y unos 75 millones de dólares a finales del 2007.

En el año 2008, Sun Microsystems adquirió MySQL por 1 millón de dólares. Dicha decisión fue muy criticada por los co-fundadores de MySQL AB, puesto que en ese momento MySQL era la tecnología que utilizaban las grandes empresas.

Debido a que la adquisición de MySQL por Sun Microsystem no fue muy fructífera, en el 2009 se llegó a un acuerdo y Oracle compró Sun Mycrosystem, junto con los derechos de autor y marca registrada MySQL. Dicha adquisición se vio atrasada debido a complicaciones legales con la Comisión Europea, pero en el año 2010 se hizo oficial.

Actualmente, la aparición del cloud computing ha hecho que MySQL se haga compatible con algunos de sus servicios, como por ejemplo Amazon EC2. Varios modelos de implementación han sido utilizados para la implementación de MySQL en plataformas de cloud computing. El modelo más popular es “imágenes en máquinas virtuales”, que permite el uso de una imagen ya hecha máquina donde esta MySQL preinstalado.

Un segundo modelo de cloud computing es gestionar MySQL cloud hosting donde la base de datos no está disponible como un servicio pero que está alojado y administrado en el nombre del propietario. Este modo, sin embargo, se ofrece sólo para algunas empresas. Con la expansión de la computación en nube y la tecnología relacionada, también se espera que las versiones de MySQL para la computación en nube puedan aumentar en número.

Información obtenida de [15].

## 2.2.6 JQuery

Fue creado inicialmente por John Resing, JQuery es la biblioteca de JavaScript más utilizada que simplifica la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas Web. Se presentó en el año 2006 en el BarCamp NYC.

Podríamos decir que es un framework de JavaScript, es decir, sirve como base para la programación avanzada aportando una serie de funciones o códigos para realizar tareas que, de otra manera requerirían de mucho más código. Por decirlo de otra forma, es una librería de código que contiene procesos o rutinas listos para usar. Suelen utilizarlo los desarrolladores sobre todo para no tener que implementar ellos mismos las tareas más básicas. Por ejemplo, con JQuery obtendremos ayuda en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax, etc. De hecho, en lo que más nos puede ayudar JQuery es al permitirnos programar sin tener que preocuparnos del navegador que esté utilizando el usuario, puesto que el código funcionará exactamente igual en todas las plataformas más habituales.

Además, con JQuery obtenemos todas estas ventajas de forma gratuita, ya que es de software libre y de código abierto. Tiene también un doble licenciamiento, uno bajo la Licencia MIT y otro para la Licencia Pública General de GNU permitiendo así su uso en proyectos tanto libres como privados.

Algunas empresas como Microsoft y Nokia han anunciado que incluirán la biblioteca en sus aplicaciones.

Por otro lado, JQuery cuenta con numerosas bibliotecas, como por ejemplo, una de las utilizadas en este proyecto, llamada **jQuery UI**. Esta biblioteca añade una serie de plugins, widgets y efectos visuales para la creación de aplicaciones Web. La biblioteca se divide en cuatro módulos, todos ellos desarrollados de acuerdo a la filosofía de JQuery:

- Núcleo: contiene las funciones imprescindibles para el resto de módulos.
- Interacciones: añade comportamientos complejos a los elementos, por ejemplo con la propiedad draggable o droppable.
- Widgets: es un conjunto de controles UI, cada uno de dichos controles tienen una serie de opciones a las que se les puede aplicar estilos con CSS y son además configurables. Un ejemplo de esto sería dialog, que es una ventana de contenidos o progressbar, que es una barra de progreso, utilizado para mostrar el tiempo de respuesta en las preguntas del trivial.
- Efectos: es una API para ayudar a los programadores a realizar interacciones y añadir transiciones animadas. Un ejemplo de esto podría ser el elemento drop o el clip.

En la actualidad, AngularJS ha empezado a desbancar a JQuery al pasar de ser una librería para convertirse en un framework de aplicaciones web. AngularJS no solo te permite una serie de funciones y mecanismos para acceder a los elementos de la página y modificarlos, sino que también te ofrece una serie de mecanismos por los cuales extender el HTML, para hacerlo más semántico, incluso ahorrarte muchas líneas de código JavaScript para hacer las mismas cosas que antes hacías con JQuery. Pero la principal diferencia y por la cual AngularJS toma la denominación de "framework", es que te marca una serie de normas y hábitos en la programación, principalmente gracias al patrón MVC implementado en AngularJS.

Información obtenida de [28].

### 2.2.7 Twitter Bootstrap

Es un framework de software libre, utilizado para dar diseño a páginas y aplicaciones Web. Fue desarrollao originalmente por ingenieros en Twitter. Contiene una serie de plantillas de diseño con tipografía, botones, formularios y otros elementos de diseño

basado en HTML, CSS y JavaScript. Es el framework más utilizado en GitHub y lo usan también organizaciones como la NASA y la MSNBC.

En el año 2011, se creó Bootstrap como una solución interna para solventar los problemas de inconsistencias en el desarrollo, dentro del equipo de ingeniería de Twitter. Puesto que no tenían establecido ninguna convención sobre las formas en las que los ingenieros de Twitter desarrollaban la plataforma.

Esto ocurría debido a que había muchos ingenieros trabajando en el mismo proyecto y cada uno tenía su propia forma de abordar los problemas, por lo que las inconsistencias eran inevitables. Además, dichas inconsistencias en el desarrollo web, pueden llevar a que el proyecto tenga incertidumbre y unos costes de mantenimiento más altos de lo normal. Por este motivo, Mark Otto y Jacob Thorton intentaron fomentar la utilización de Bootstrap y consiguieron que de esta manera, el equipo trabajara más rápido, de forma más eficaz y con menos inconsistencias.

A pesar de que fue desarrollado como una solución interna en Twitter, Marck y Jacob, pensaron que el proyecto tenía un gran potencial y en el año 2011 Bootstrap fue publicado en Github como un proyecto de código abierto. Poco después se convirtió en el proyecto de código abierto más activo del mundo al contribuir en el proyecto miles de desarrolladores de todo el mundo. Desde entonces, Bootstrap ha ganado más notoriedad y en la actualidad es uno de los framework de presentación más utilizado, sobre todo a la hora de desarrollar proyectos responsive y para móviles en Internet.

Bootstrap tiene una serie de elementos personalizables de HTML, CSS y JavaScript. Además, gracias a que es de código abierto, está en una mejora continua, se le han añadido una serie de funcionalidades, como responsividad 100% a dispositivos móviles y algunos plugins de jQuery.

Unos años después, en el 2013, salió Bootstrap 3 con un nuevo diseño, mejor gestión de errores y mejores opciones de personalización entre otras mejoras, todo para hacer más fácil la tarea de los desarrolladores.

Información obtenida de [14].

## 2.3 Páginas web educativas

La tecnología ha crecido enormemente en los últimos años en nuestra sociedad y los niños la utilizan cada vez a una edad más temprana. Hay diferentes opiniones sobre si esto es beneficioso o no para los niños. Por un lado, es bueno que los niños tengan experiencias reales con objetos, pero por otro, tampoco se ha demostrado que la utilización de la tecnología en los niños tenga un efecto negativo.

Actualmente, hay una gran cantidad de páginas web educativas en internet, a continuación veremos algunas de ellas parecidas al “Mini trivial”.

### 2.3.1 Trivinet

Trivinet nace de las manos de un docente que trata de desarrollar una herramienta, un recurso didáctico, que usar con su propio alumnado de modo que consiga que aprendan mientras juegan.

Lo cierto es que trivinet va mucho más allá. El docente puede detectar deficiencias en el aprendizaje conforme recibe las preguntas que sus alumnos sugieren a través de la herramienta y gracias a esta detección temprana puede corregirlo. Además consigue que el alumnado repase la teoría en el momento en que sugiere preguntas. Por último, consigue que los alumnos sean capaces de memorizar los conceptos que se tratan en el juego de trivial.

Trivinet no es como trivial pursuit en el sentido de jugar con “quesitos” sino que se trata de un juego de preguntas y respuestas de muchas y variadas temáticas y con distintas dificultades para cada pregunta.

Este recurso didáctico gratuito se encuentra en uso a día de hoy en muchos niveles educativos y en varios países con mucho éxito.

La herramienta también permite a los docentes generar exámenes tipo test a partir de las preguntas que forman el grupo.

## Jugar al trivial online

¿Cuál no es un mecanismo de destrucción de la célula diana en la hipersensibilidad citotóxica?

4

Detener

🔄

🔄

Fagocitosis de las estructuras diana

Activación del complemento


Mecanismo de citotoxicidad

Efecto citotóxico

Esta pregunta se ha jugado 283 veces y se ha acertado en 65 ocasiones ( 22,97% de acierto)

Esta pregunta pertenece a Biología su dificultad es Muy difícil

Guarda el enlace posterior si desea comenzar a jugar al trivial online desde esta pregunta: ¿Cuál no es un mecanismo de destrucción de la célula diana en la hipersensibilidad citotóxica?

Esta pregunta se modificó el 11-05-2015y fué enviada inicialmente por  belenwi

Si crees que existe algún error en esta pregunta puedes indicarlo en el foro indicando el identificador de la pregunta:3572

[Jugar preguntas de Biología](#)

[Jugar preguntas muy difíciles](#)

### Estadísticas de la partida

Bienvenido usuario no registrado

Hoy N° aciertos 0

Hoy N° respuestas 0

 Registrarte Accede gratis a las ventajas

### Uso de cookies

trivinet.com utiliza cookies, tanto propias como de terceros.

Si continúa navegando consideramos que acepta su uso

Aceptar

[Leer mas](#)

@Pablo Ruiz Soria - TriviNet.com - Juego de trivial online - Version web

[Política de privacidad](#) - [Contacto](#) - [Acerca de](#) - [Videotutoriales](#) - [Aviso legal](#)

### Ilustración 4: trivinet

Información obtenida de [5].

## 2.3.2 Trivial Pursuit

Este conocidísimo juego de mesa fue inventado por Scott Abbott y Chris Haney en 1979 y no fue hasta el año 1981 cuando por fin se lanzó al mercado. Hasta el año 1984 no se hizo popular en Estados Unidos y se vendieron unos 20 millones de ejemplares. Más tarde, en 1988, Parker Brothers compró los derechos del juego y hasta el año 2004 se vendieron cerca de 88 millones de ejemplares en 26 países y 17 idiomas.

El juego consiste en ir respondiendo a una serie de preguntas para poder obtener las cuñas de puntuación (quesitos). Los jugadores empiezan en el centro del tablero y según van tirando los dados (en orden), van moviendo sus fichas en la dirección elegida. Al acertar una pregunta, el jugador puede volver a tirar los dados y si no la responde correctamente, será el turno del siguiente jugador. Hay una serie de casillas especiales, situadas en las intersecciones, si el jugador acierta una pregunta situado sobre una de estas casillas, obtendrá como premio una de las seis cuñas del color de la casilla. Además cada casilla tiene un color, que corresponde a un tema determinado: geografía, espectáculo, historia, arte y literatura, ciencias y naturaleza o deportes y pasatiempos.





Ilustración 5: Trivial pursuit

El trivial ha pasado de ser un juego de mesa a estar desarrollado para una gran cantidad de plataformas, como puede ser el PC, videoconsolas y móviles.

Información obtenida de [6].

### 2.3.3 Testeando

Esta herramienta web se basa en un juego de preguntas respuestas tipo test o trivial. Su objetivo es ayudar a los profesores y a los alumnos afianzando los conocimientos adquiridos en el colegio y enseñando a los alumnos conceptos nuevos.

La web está dividida en diversos cursos comprendidos entre 1º de primaria y 2º de bachillerato. Lo primero que debes hacer es elegir un curso, a continuación deberá seleccionar una de las asignaturas de dicho curso, hecho esto, se accederá a otra ventana en la cual podremos elegir uno de los temas existentes y empezar así a jugar.

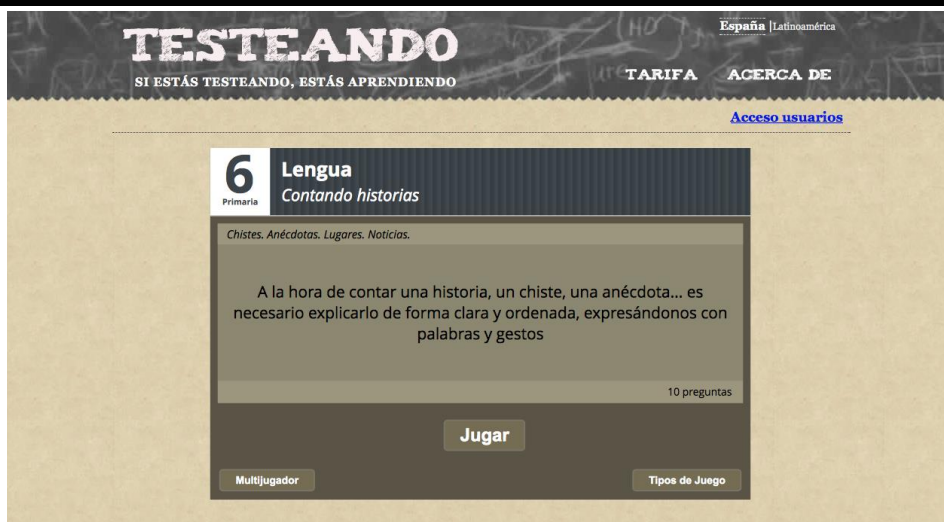


Ilustración 6: Testeando

El juego consta de una pregunta y 4 posibles respuestas, de las cuales solo una es correcta. Las preguntas respondidas correctamente puntúan positivamente teniendo en cuenta además el tiempo que el jugador ha tardado en responder dicha pregunta. Por otro lado, las preguntas falladas, restan puntos del resultado final. Además, durante el transcurso del juego, el jugador podrá hacer uso de hasta tres comodines que le ayudarán a la correcta contestación de las preguntas.

Una vez finalizada la partida, el jugador (alumno) podrá elegir entre repetir el test con exactamente las mismas preguntas para repasar así conceptos o comenzar otro con preguntas diferentes.

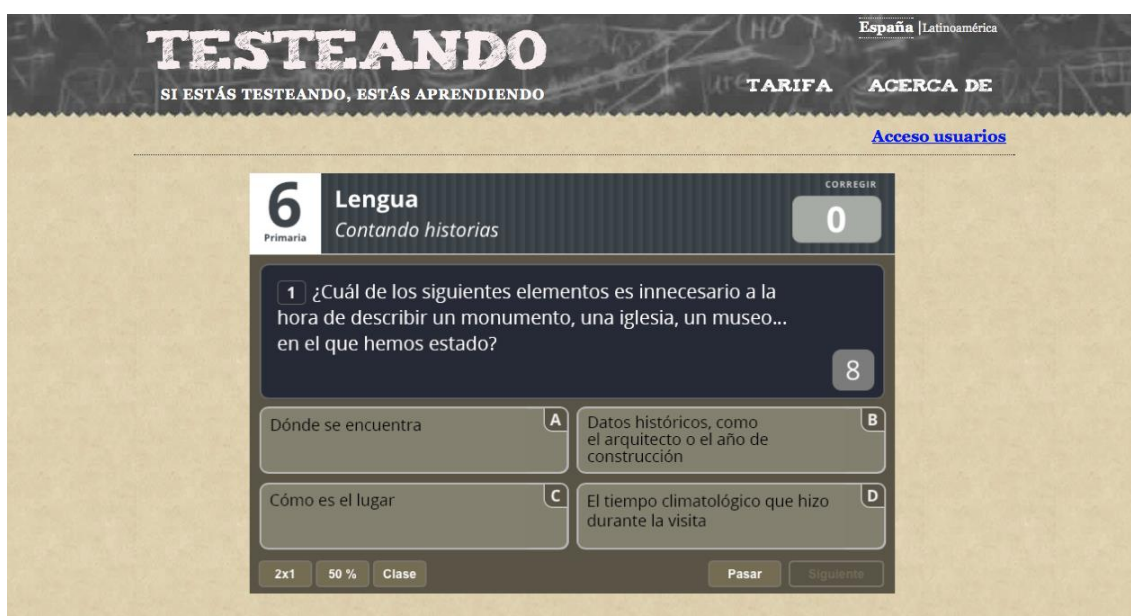


Ilustración 7: testeando, juego

Como podemos observar en la imagen de arriba, la pantalla se puede dividir en 4 partes:

- Parte superior de la pantalla: aquí se puede visualizar el curso, materia y tema elegido, así como la puntuación que llevamos acumulada.
- Parte central superior: aquí podemos ver el número de la pregunta en la cual nos encontramos (normalmente serán 10), el texto de la pregunta y los puntos ganados o perdidos dependiendo de si contestamos correctamente o no a la pregunta.
- Parte central inferior: en esta parte se encuentran las respuestas, al pulsar sobre una ésta cambiará de color. Tendrá un fondo verde si la respuesta es la correcta o rojo en caso contrario.
- Parte inferior: en la parte de la izquierda se encuentran los diferentes comodines y en la parte de la derecha está el botón para seguir con la siguiente pregunta.

Esta web ofrece además varias modalidades de juego, clásico, trípex, infinitum... con diferentes reglas para que los usuarios puedan elegir la más adecuada para cada momento. Además la web ofrece dos tipos de versiones, una LITE (gratuita) y otra PRO (con suscripción de pago). Cuya diferencia se encuentra principalmente en el número máximo de partidas y conexiones (ordenadores) simultáneas permitidas.

Información obtenida de [7].

### 2.3.4 Triviados

Triviados es un juego online basado en el popular juego de mesa de preguntas y respuestas Trivial Pursuit. Enfrenta a dos usuarios a resolver múltiples preguntas de varias categorías. Está disponible de modo gratuito para todo tipo de smartphones, tablets, ordenadores de sobremesa y portátiles.

Fue creado a finales de 2012 por la firma Pandereta Estudio. Ante el enorme éxito de descargas que obtuvo la aplicación, sus tres cofundadores, Francisco Calvelo, Álvaro Urbano e Iván Rubio se plantearon realizar concursos y premiar a los mejores usuarios que ganaran más partidas. Mediapro y Pandereta Estudio llegaron a un acuerdo en julio de 2013 para adaptar Triviados al formato televisivo.

Las normas y el funcionamiento de Triviados están basadas en el juego Trivial Pursuit. Los dos contrincantes deberán conseguir siete estrellas respondiendo acertadamente a las siete preguntas de la categoría correspondiente. Para ello deberán recorrer todo el

tablero, utilizando un dado, con el fin de caer en las casillas especiales de cada categoría.

Si la respuesta a una pregunta es errónea, se perderá el turno, hasta que el contrincante también se equivoque. Entre las casillas de pregunta de estrella, hay seis casillas de distancia. Dos de esas casillas permiten volver a tirar con el dado. En el tablero hay dos casillas que permiten retar al contrincante. Para ello, los dos usuarios responderán a una ronda de siete preguntas de las diferentes categorías. El que acierte más preguntas ganará el reto, mientras que el otro usuario perderá una estrella. Si el reto termina en empate, ninguno de los dos perderá estrellas.

El juego termina cuando uno de los dos usuarios consigue responder a las preguntas de estrellas de cada categoría, y a continuación responde correctamente un mínimo de cuatro preguntas de la ronda final. En esta ronda, se formulan siete cuestiones al azar de las siete categorías existentes.

Al ser un juego online, permite enfrentar a diferentes oponentes, usuarios de Triviados. La forma más común de iniciar el juego es creando una partida con un oponente amigo, al que se ha agregado previamente. También se puede iniciar una partida con un usuario aleatorio, o buscarlo en la red social Facebook. Por último, se puede jugar en una partida múltiple con diferentes contrincantes.



Ilustración 8: Triviados

Información obtenida de [29].

### 2.3.5 Comparación de resultados

A continuación analizaremos las diferencias de cada una de las web vistas anteriormente y las compararemos con “Mini trivial”.

Aplicaciones	Trivinet	Trivial Pursuit	Testeando	Triviados	Mini trivial
Objetivos					
Web educativa	SI	SI	SI	NO	SI
Online	SI	SI	SI	SI	SI
Gratuito	SI	NO	SI (*)	SI	SI
Interfaz llamativa	SI	SI	SI	SI	SI
Foro	SI	NO	NO	SI	NO

<b>Editable</b>	SI(**)	NO	NO	SI(**)	NO
<b>Clasificación por temas</b>	SI	SI	SI	SI	SI
<b>Modos de pregunta</b>	NO	NO	SI	NO(***)	SI
<b>Baja complejidad</b>	SI	SI	SI	SI	SI
<b>Ameno</b>	SI	SI	SI	SI	SI

(\*) Tiene una versión gratuita (LITE) pero también tiene otra de pago (PRO).

(\*\*) Se pueden sugerir nuevas preguntas o comunicar si en alguna de ellas ha habido algún error.

(\*\*\*) A pesar de que tiene un modo reto contra otro oponente, en dicho modo, no se cambia el formato de las preguntas y las respuestas.

Como puede observarse en la tabla anterior, “Mini trivial” se asimila bastante a las aplicaciones que hay en el mercado. Sin embargo, tiene una funcionalidad que la diferencia bastante del resto puesto que tiene diferentes tipos de preguntas/respuestas.

Por otro lado, algunas de sus futuras mejoras serían la implementación de un foro para que los distintos usuarios puedan interactuar unos con otros y añadir un apartado que permita a los usuarios el envío de sugerencias tanto de nuevas preguntas como de preguntas que contienen algún error y deben ser corregidas.

## 2.4 Marco regulador

A la hora de crear una página web, debes tener en cuenta la normativa vigente, sobre todo en los casos de páginas web corporativas o de páginas web que ofrecen sus productos y servicios online.

De entre todas las normativas que pueden afectar a una página web, es preciso prestar especial atención a las siguientes:

- Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico (LSSICE).
- Ley de Protección de Datos de carácter personal (LOPD).

Además, también existen otras regulaciones sobre la defensa de los consumidores, el comercio minorista, la propiedad intelectual, etc.

Uno de los aspectos más importantes de la LSSICE es algo tan sencillo como ser conscientes de que toda empresa tiene unos datos identificativos que deben constar en una página web: nombre o denominación social, número de identificación fiscal, datos de inscripción en el Registro Mercantil, medios de contacto, datos correspondientes a nuestra profesión, referencias a precios e impuestos, actividad, etc.


Por supuesto, cuanta más información demos de nuestro negocio en nuestra página web, más confianza generaremos en los usuarios. Especialmente si comercializamos nuestros servicios a través de Internet.

Por otro lado, también debemos ser conscientes de que la recogida y tratamiento de datos de carácter personal (nombre y apellidos, DNI, dirección postal, etc.) debe realizarse siempre recabando el consentimiento del cliente y cumpliendo con lo establecido en la LOPD y su normativa de desarrollo. Por tanto, debe tenerse en cuenta la protección de datos de carácter personal de los usuarios de su página web.

Asimismo, es preciso cumplir con lo establecido en la normativa vigente en materia de consumidores y usuarios, con el fin de respetar los derechos de éstos.

Otro aspecto importante es ser consciente de que se deben respetar los derechos de propiedad intelectual e industrial de terceras partes. Una autorización previa al incluir logotipos, firmas o marcas de terceros (fotografías, escritos, vídeos, etc.) respetando sus indicaciones será suficiente para tratar debidamente un material susceptible de





este tipo de derechos. Sin embargo, al no desarrollar este proyecto con un fin lucrativo, no es necesario dicha autorización.

Además no es lo mismo que la web esté dirigida a empresas, a usuarios finales o, por ejemplo, a menores de edad. En cada caso las obligaciones legales de tu página web pueden cambiar. Este proyecto ha sido adaptado a un uso de usuarios de todo tipo de edades debido a su contenido educativo.

Información obtenida de [30].



### 3 Descripción del problema

Hoy en día, algunos niños no obtienen un rendimiento académico adecuado, lo cual supone una preocupación para las familias y acerca del propio futuro del alumno. Este fenómeno es llamado “fracaso escolar”.

El fracaso escolar no responde a una única causa, cada alumno tiene un caso particular y su nivel de rendimiento académico puede estar determinado por infinidad de factores. A continuación analizaremos las principales causas de este fenómeno:

- Dificultades propias del estudiante.

Un elevado porcentaje de casos de fracaso escolar se debe directamente a dificultades como la dislexia, el déficit de atención con hiperactividad (TDAH) o problemas de audición o de visión. Es muy importante detectar dichas causas a tiempo para que el alumno no pierda el ritmo de aprendizaje de la clase y para no dar diagnósticos erróneos como “eres un vago” o “no vales para estudiar”.

Por otra parte, al igual que los adultos, los niños necesitan tener una motivación que les impulse a esforzarse. Si se aburren o no están motivados, no se esforzarán ni verán el sentido de las clases. Un ejemplo de ello podría ser el de los alumnos con un alto coeficiente intelectual, que debido a que su capacidad va por delante del nivel que se da en la clase, pierden rápidamente el interés. Además durante la adolescencia, algunos niños suelen experimentar una baja autoestima o depresiones, factores que también afectan de forma importante en el rendimiento del alumno.

- Fallos en el sistema educativo.

España es uno de los países de la Unión Europea con una de las mayores tasas de fracaso escolar y tiene la mayor tasa de abandono, con un 24,99%. Reducir el fracaso escolar ha sido siempre uno de los objetivos de los sucesivos gobiernos, que han ido probando distintas leyes de reforma del sistema educativo.

La educación reglada, es una inmensa cadena en la que cada eslabón importa y uno de los eslabones más importantes son los profesores. Un profesor que tenga amplios conocimientos de la materia que imparte y que además sea capaz de motivar e implicar a su aula, puede ayudar a los más rezagados a reengancharse al ritmo del resto y aumentar su autoestima.

- Causas socioeconómicas.

Algunos estudios revelan que hay una relación directa entre el rendimiento de los hijos en los colegios y el nivel socioeconómico de las familias. Casos en los cuales las familias no tienen el suficiente dinero para que los hijos continúen los estudios o familias que dan poco apoyo a los niños en casa, provocan que los niños no den lo mejor de sí o que abandonen su etapa formativa antes de tiempo. Dichos colectivos de alumnos son los que más dependen de la inversión pública en educación para conseguir buenos rendimientos.

Por eso el papel de la familia es muy importante y los valores y modelos que les transmitan a los niños serán cruciales en sus intereses, hábitos y forma de pensar.

- Malestar emocional.

Cualquier problema que les provoque un malestar emocional, como baches anímicos o incluso situaciones de acoso escolar, va a influir en su rendimiento académico. Unas veces no son más que periodos pasajeros, pero otras marcan más de lo deseable el aprendizaje.

- Causas pedagógicas.

Estas causas hacen referencia a las técnicas y el hábito de estudio, causantes en algunas ocasiones del bajo rendimiento. Si no enseñamos a los niños a estudiar, será difícil que obtengan los resultados deseados.

Este proyecto ha sido desarrollado como herramienta de apoyo para solventar los problemas del fracaso escolar. Gracias a “Mini trivial” podrán repasar conocimientos adquiridos en clase, ejercitando así su memoria, puesto que esta capacidad es clave para conseguir un buen rendimiento académico. El uso de “Mini trivial” ayudará también en la motivación de los niños, dando con el juego una imagen positiva de lo académico.

Información obtenida de [16], [17] y [18].

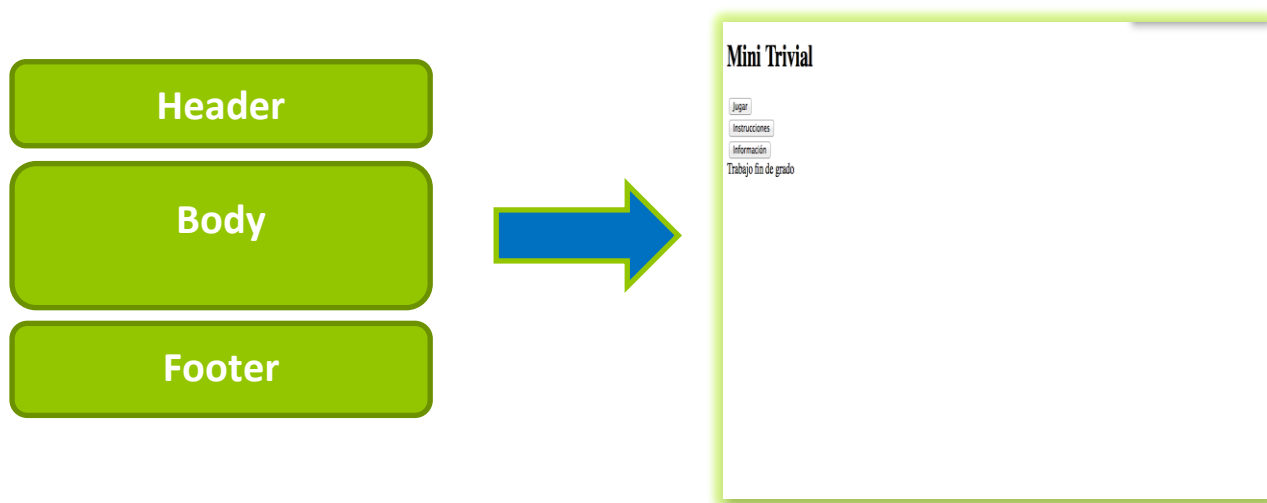
## 4 Diseño y desarrollo de la solución

En el apartado 2 han sido explicadas la mayoría de las tecnologías aplicadas en este proyecto a modo de introducción y para un mejor entendimiento. A continuación, explicaré el modo de desarrollo de esta aplicación.

### 4.1 Implementación HTML

Lo primero que se hizo fue descargar un editor para poder desarrollar el código tanto HTML como CSS o JavaScript. Después de buscar y comparar los diferentes editores del mercado, se decidió utilizar Bluefish, puesto que es una herramienta gratuita y con unas buenas críticas.

Inicialmente se crearon una serie de pantallas mediante HTML, todas ellas con la misma estructura y sin ninguna regla de estilo, tal y como puede observarse en la figura situada debajo.



## 4.2 Implementación CSS

Después se implementó un documento .css para dar forma a las páginas HTML y conseguir así un aspecto mejor. Hay 3 formas distintas de incluir código CSS dentro de una página HTML:

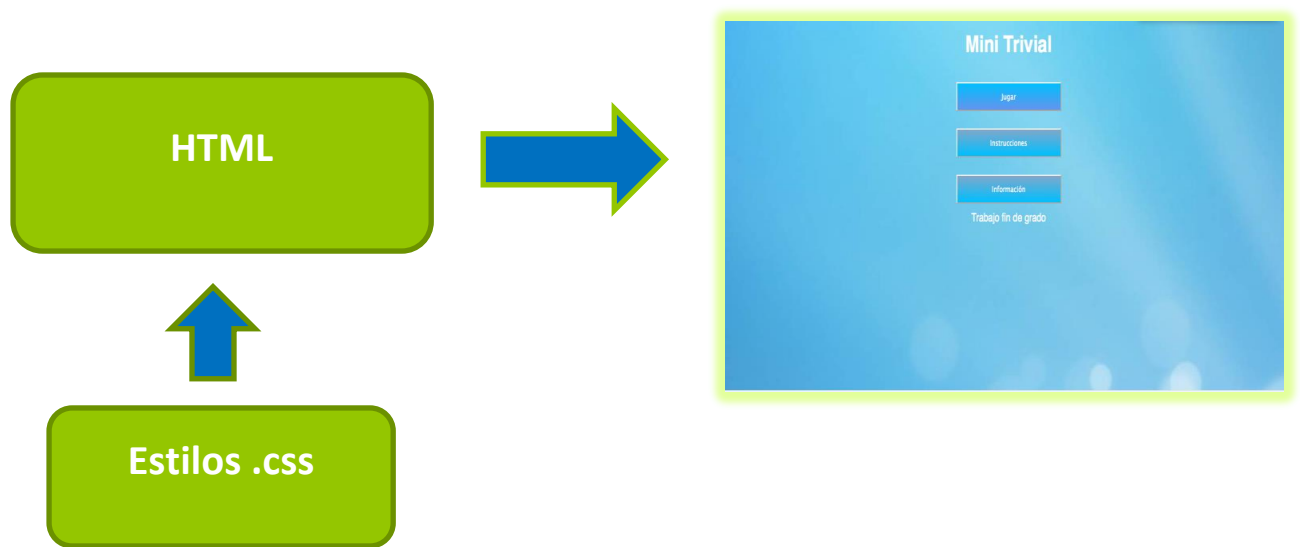
- Incluirlo en el propio documento HTML. Esto se consigue incluyendo la etiqueta `<style>` dentro de la cabecera del documento HTML y escribiendo dentro los estilos que queramos incluir. Es importante definir en dicha etiqueta el atributo `type="text/css"`.

Su principal desventaja es que si queremos cambiar el aspecto de todas las páginas que tengamos, tendremos que cambiar los estilos de cada una de dichas páginas una a una.

- Definir CSS en un archivo externo. De esta forma, se incluyen mediante la etiqueta `<link>` los archivos con extensión .css a las páginas HTML. Pero para que funcione, debemos incluir una serie de atributos a dicha etiqueta:
  - rel: indica el tipo de relación que tiene el recurso enlazado y la página HTML. En CSS, se usa `stylesheet`.
  - type: indica el tipo de recurso enlazado. En CSS, es `text/css`.
  - href: indica la URL del archivo CSS que contiene los estilos. La URL puede ser relativa o absoluta.
  - media: indica el medio en el que se van a aplicar los estilos del archivo CSS. En principio, para este ejemplo no es importante, lo normal es poner `all`.
- Incluir CSS en los elementos HTML. Consiste en poner la etiqueta `<style>` como atributo de la etiqueta que queremos personalizar.

En este proyecto, se ha intentado incluir siempre los estilos en archivos externos, sin embargo, en algunas ocasiones, también se ha utilizado esta última opción, añadiendo algún estilo dentro de la etiqueta HTML.

Una vez añadido el código CSS, la aplicación quedó de la siguiente forma:



Más tarde, se implementaron varios estilos con las librerías de Bootstrap para dar uniformidad a la Web.

## 4.3 Implementación JavaScript

Ahora que ya está la página web lista, es hora de incluir funcionalidades con JavaScript. La integración de JavaScript y HTML es muy flexible, ya que existen al menos tres formas de incluir código JavaScript en las páginas web.

- Incluir JavaScript en el mismo documento HTML. El código JavaScript se encierra entre etiquetas `<script>` y se incluye en cualquier parte del documento. Aunque es correcto incluir cualquier bloque de código en cualquier zona de la página, se recomienda definir el código JavaScript dentro de la cabecera del documento (dentro de la etiqueta `<head>`)

Para que la página XHTML resultante sea válida, es necesario añadir el atributo `type` a la etiqueta `<script>`. Los valores que se incluyen en el atributo `type` están estandarizados y para el caso de JavaScript, el valor correcto es `text/JavaScript`. Este método se emplea cuando se define un bloque pequeño de código o cuando se quieren incluir instrucciones específicas en un determinado documento HTML que completen las instrucciones y funciones que se incluyen por defecto en todos los documentos del sitio web.

El principal inconveniente es que si se quiere hacer una modificación en el bloque de código, es necesario modificar todas las páginas que incluyen ese mismo bloque de código JavaScript.

- Definir JavaScript en un archivo externo. Las instrucciones JavaScript se pueden incluir en un archivo externo de tipo JavaScript que los documentos XHTML enlazan mediante la etiqueta `<script>`. Se pueden crear todos los archivos JavaScript que sean necesarios y cada documento XHTML puede enlazar tantos archivos JavaScript como necesite.

Además del atributo `type`, este método requiere definir el atributo `src`, que es el que indica la URL correspondiente al archivo JavaScript que se quiere enlazar. Cada etiqueta `<script>` solamente puede enlazar un único archivo, pero en una misma página se pueden incluir tantas etiquetas `<script>` como sean necesarias.

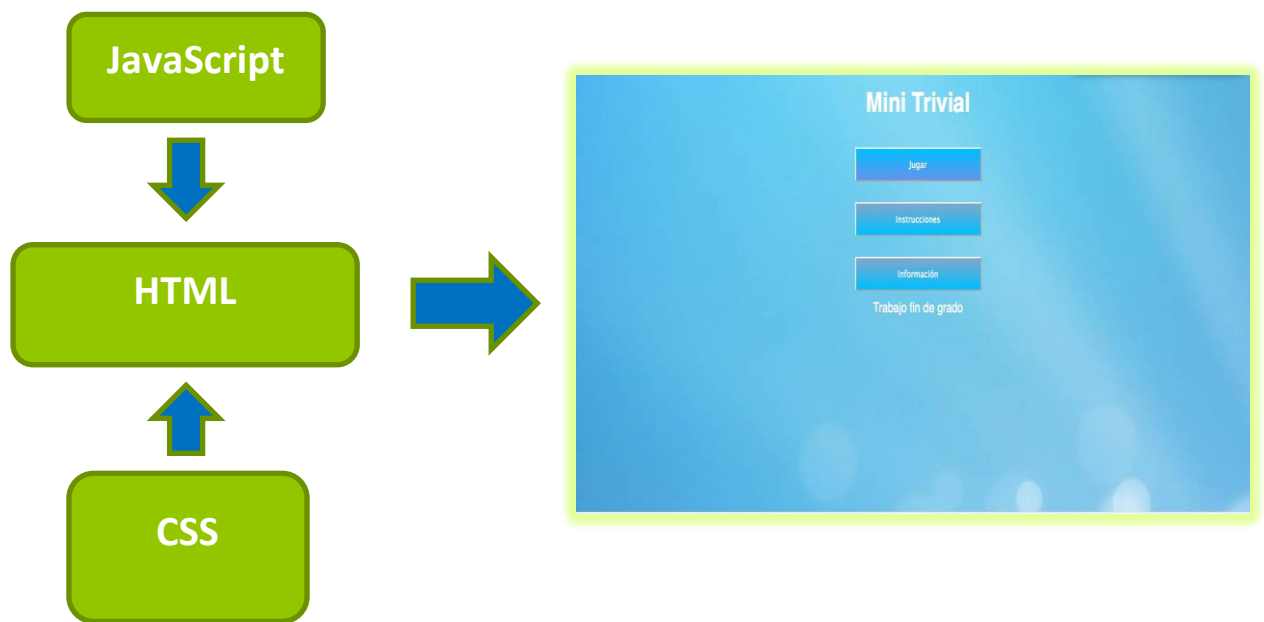
Los archivos de tipo JavaScript son documentos normales de texto con la extensión `.js`, que se pueden crear con cualquier editor de texto como Notepad, Wordpad, EmEditor, UltraEdit, Vi, etc.

La principal ventaja de enlazar un archivo JavaScript externo es que se simplifica el código XHTML de la página, que se puede reutilizar el mismo código JavaScript en todas las páginas del sitio web y que cualquier modificación realizada en el archivo JavaScript se ve reflejada inmediatamente en todas las páginas XHTML que lo enlazan.

- Incluir JavaScript en los elementos HTML. Este último método es el menos utilizado, ya que consiste en incluir trozos de JavaScript dentro del código XHTML de la página. El mayor inconveniente de este método es que ensucia innecesariamente el código HTML de la página y complica el mantenimiento del código JavaScript. En general, este método sólo se utiliza para definir algunos eventos y en algunos otros casos especiales, como se verá más adelante.

Se intentó incluir siempre el lenguaje JavaScript en archivos externos, sin embargo, en algunas ocasiones y por comodidad, también se ha incluido dentro de algunas etiquetas HTML, como por ejemplo las acciones que capturan el evento “onclick” de la mayoría de los botones.

Al añadir el código JavaScript, la aplicación web se volvió más dinámica. Entre otras cosas, ahora al pulsar sobre los botones se podía ir accediendo a las diferentes pantallas HTML.



## 4.4 Implementación JQuery y MySQL

Conseguida una página web dinámica y funcional, el siguiente paso fue lograr un “trivial básico”, que únicamente tuviera la funcionalidad de mostrar una pregunta y sus cuatro respuestas. Al principio, las preguntas estaban almacenadas en un Json al que se accedía mediante JavaScript y se representaban las diferentes preguntas y respuestas.

Más tarde, se cambió la forma de obtener y mostrarlas preguntas y respuestas, gracias a jQuery se podían añadir elementos HTML al DOM dentro del archivo .js. De esta forma, se creó un elemento <h3> para representar cada una de las preguntas y cuatro elementos <input> para los botones de las 4 posibles respuesta y mediante jQuery los añadió al DOM. Cuando se pasa de una pregunta a la siguiente, se borran dichos elementos y se crean los nuevos, es decir, se crea la siguiente pregunta/respuestas. De esta manera se consiguió implementar un trivial básico al que se fue añadiendo complejidad de forma progresiva.

Por otro lado, el acceso a las preguntas y respuestas almacenadas en la base de datos MySQL se hace también mediante una combinación de jQuery y PHP. El documento

PHP realiza la conexión con la base de datos, llamada “trivial”, para posteriormente acceder a la tabla “preguntas”. Como su propio nombre indica, de esta tabla y de forma aleatoria, se obtienen las preguntas que se mostrarán posteriormente en la aplicación. El campo id de la tabla preguntas está relacionado con el campo preguntas de la tabla respuesta, de esta forma se puede recuperar las 4 respuestas correspondientes a una pregunta.

A continuación se muestra cómo está estructurada la base de datos:

Base de datos	Tabla	Campo	Funcionalidad
trivial	preguntas	id	Campo identificativo
		texto	Es el texto de la pregunta
		text	Es el texto de la pregunta tras seleccionar el idioma inglés.
		imagen	Es la ruta a una imagen guardada en el servidor
		tema	Es el tema de la pregunta
		tipo	Es el tipo de pregunta, se verá más adelante los posibles tipos de preguntas.
		puntos	Son los puntos que obtendrá el usuario si responde correctamente a la pregunta
	respuestas	id	Campo identificativo
		pregunta	Clave ajena. Cada respuesta se relaciona con una pregunta.
		texto	Contiene el texto de las respuestas
		text	Contiene el texto de las respuestas en inglés
		esCorrecto	Puede tener dos valores, 1 si la respuestas es la correcta o 0 si no lo es
	encuesta_preguntas	id	Campo identificativo
		descripcion	Es la descripción de la pregunta de la encuesta
	encuesta_respuestas	ip	Es el ip del usuario que realiza la encuesta
		puntuación	Es la puntuación elegida por un usuario
		id	Clave ajena. Cada pregunta se relaciona con una puntuación
	puntuaciones	id	Campo identificativo
		nombre	Es el nombre que el usuario introduce para guardar su puntuación
		puntos	Es la puntuación obtenida por un usuario



Se añadió posteriormente las puntuaciones, implementado en un principio mediante localStorage. Haremos una pequeña pausa para un mejor entendimiento de esta nueva tecnología.

LocalStorage es un espacio de almacenamiento local y tienen una funcionalidad muy parecida a la de las conocidas cookies. Cuenta además con varias ventajas:

- Tiene un espacio menos limitado que las cookies, mientras que estas últimas pueden almacenar hasta 4KB de espacio, en localStorage se puede almacenar entre 5 y 10 MB dependiendo del navegador.
- La información almacenada con localStorage no es enviada al servidor con cada petición, al contrario de lo ocurrido con las cookies. Que cada vez que realizan una petición al servidor, toda la información que es almacenada es enviada y también es recibida nuevamente con la respuesta del servidor.
- No existe una caducidad para localStorage, la información quedará almacenada hasta que se elimine expresamente. Por otro lado, las cookies tienen una caducidad.

Al guardar las puntuaciones con localStorage, éstas permanecían guardadas en el navegador, por lo que desaparecían al cambiar el navegador utilizado. Además, dichas puntuaciones eran vistas únicamente por el usuario que las guardaba y ningún otro usuario podía verlas a menos que utilizaran el mismo equipo y navegador. Se perdía así la competitividad del juego.

Por estos motivos, se decidió cambiarlo y guardar las puntuaciones mediante PHP en una base de datos. De esta manera, el documento PHP hace una llamada a una tabla de la base de datos llamada puntuaciones y en dicha tabla se van guardando todas las puntuaciones de todos los usuarios de la aplicación junto con el nombre que dicho usuario introduzca.

Se implementó también una pequeña encuesta, cuyos resultados serán comentados en el apartado 8.1. En dicha encuesta, al igual que en las puntuaciones, se guardan una serie de datos, las repuestas seleccionadas por los usuarios, en la base de datos MySQL.

Información obtenida de [27].

## 4.5 Tipos de preguntas

Una vez terminada toda la implementación del trivial básico, se comenzó a añadir tipos de preguntas-respuestas. Los tipos que se desarrollaron fueron los siguientes:

- Tipo 1.  
La pregunta es un texto simple y puede estar reforzada mediante una fotografía. Las respuestas están formadas mediante una serie de botones conteniendo en su interior texto.
- Tipo 2.  
La pregunta es un texto simple y las respuestas son 4 fotografías, cada una de ellas ampliables, con un botón de check a su izquierda con el que poder elegir una de las respuestas.
- Tipo 3.  
Lo primero que se muestra en este tipo de preguntas es un archivo de vídeo, que el usuario puede controlar a su gusto, pararlo, pausarlo, etc. Después de ver dicho vídeo, el usuario deberá pulsar el botón “Continuar” para poder visualizar la pregunta, que será un texto cuya respuesta habrá sido mencionada en el video anterior y deberá seleccionar una de las 4 posibles respuestas de texto.
- Tipo 4.  
En este tipo no es preciso responder a ninguna respuesta, sino resolver un puzle. El usuario debe colocar una serie de piezas para formar una fotografía. En la parte superior, aparecerá un texto informativo acerca de dicha fotografía que estará relacionada con el tema en el que nos encontremos.

## 5 Organización del proyecto

### 5.1 Planificación del proyecto

A continuación se explicará cómo se ha organizado el tiempo disponible para el desarrollo del proyecto, es decir, la planificación inicial estimada y la que finalmente ha sido. Además se comentarán las diferencias entre ambas planificaciones así como el método de control escogido para evitar posibles desviaciones de tiempo y recursos.

### 5.2 Planificación inicial

En la tabla de abajo se muestra el cronograma de actividades inicial. Dicho cronograma ha sido utilizado para conocer el momento en el que cada actividad se debe llevar a cabo, las tareas que ya se han completado así como la secuencia en que cada fase tiene que ser ejecutada.

Actividad	Fecha de inicio	Duración (días)	Fecha de fin
Consolidar la idea	25/01/2015	10	04/02/2015
Organización del proyecto	04/02/2015	7	11/02/2015
Desarrollo de una página web con HTML	11/02/2015	15	26/02/2015
Desarrollo de un trivial básico	26/02/2015	30	28/03/2015
Añadir diseño con CSS	28/03/2015	7	04/04/2015
Añadir tipos de preguntas	04/04/2015	20	24/04/2015
Implementación de mejoras	24/04/2015	15	09/05/2015
Pruebas finales	09/05/2015	4	13/05/2015
Conclusiones	13/05/2015	3	16/05/2015
Realización de la memoria	16/05/2015	30	15/06/2015

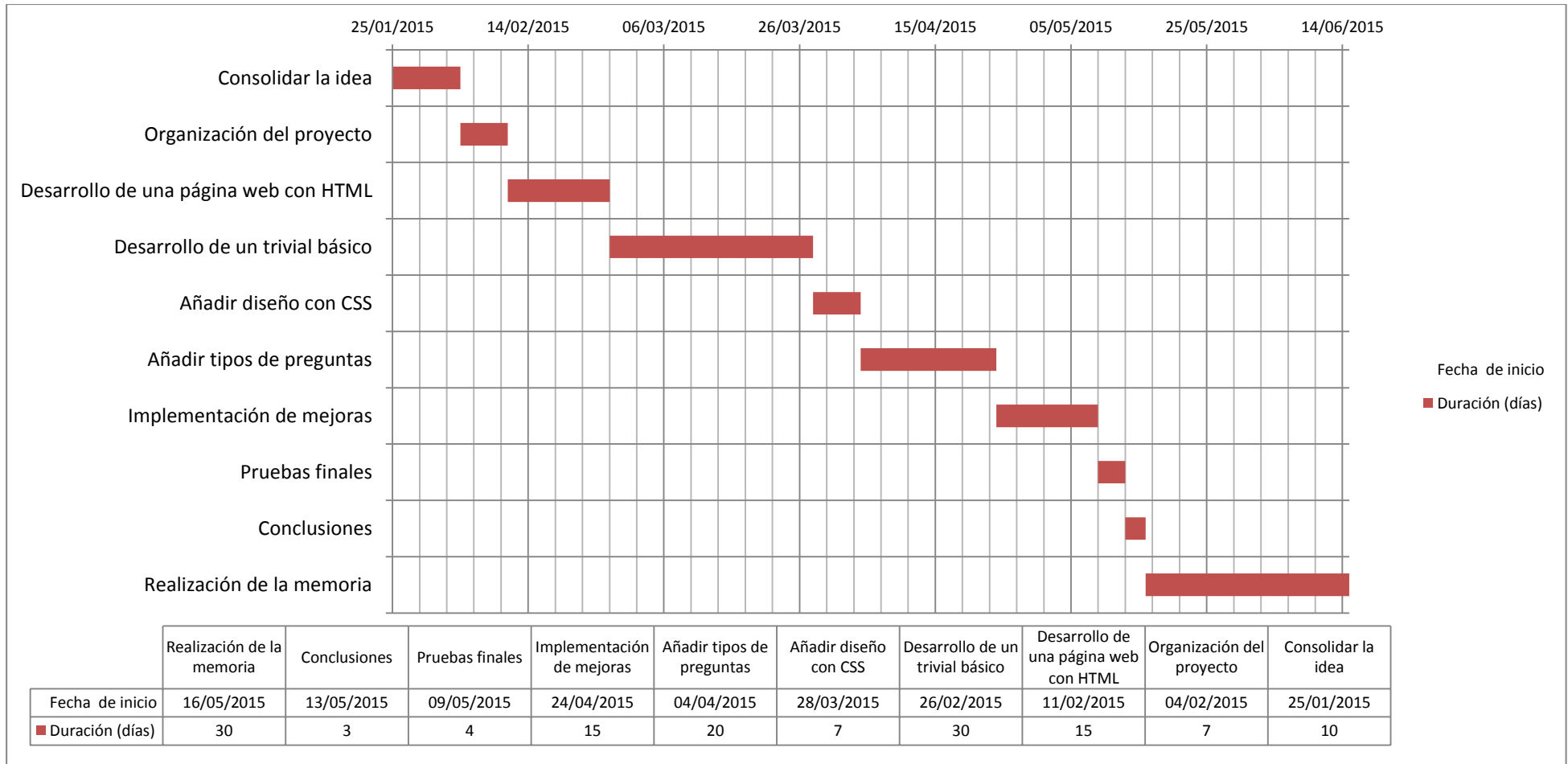


Diagrama de Gantt inicial

## 5.3 Planificación final

Al asignarse el proyecto a finales de enero, la planificación original fue pensada para entregarse en junio. Sin embargo, debido a problemas de trabajo y prácticas, se tuvo que posponer la entrega a septiembre ya que no tenía tiempo suficiente para poder dedicar al proyecto.

Actividad	Fecha de inicio	Duración (días)	Fecha de fin
Consolidar la idea	25/01/2015	20	14/02/2015
Organización del proyecto	14/02/2015	10	24/02/2015
Desarrollo de una página web con HTML	24/02/2015	15	11/03/2015
Desarrollo de un trivial básico	11/03/2015	40	20/04/2015
Añadir diseño con CSS	20/04/2015	7	27/04/2015
Añadir tipos de preguntas	27/04/2015	30	27/05/2015
Implementación de mejoras	27/05/2015	20	16/06/2015
Pruebas finales	16/06/2015	7	23/06/2015
Conclusiones	23/06/2015	6	29/06/2015
Realización de la memoria	29/06/2015	45	13/08/2015

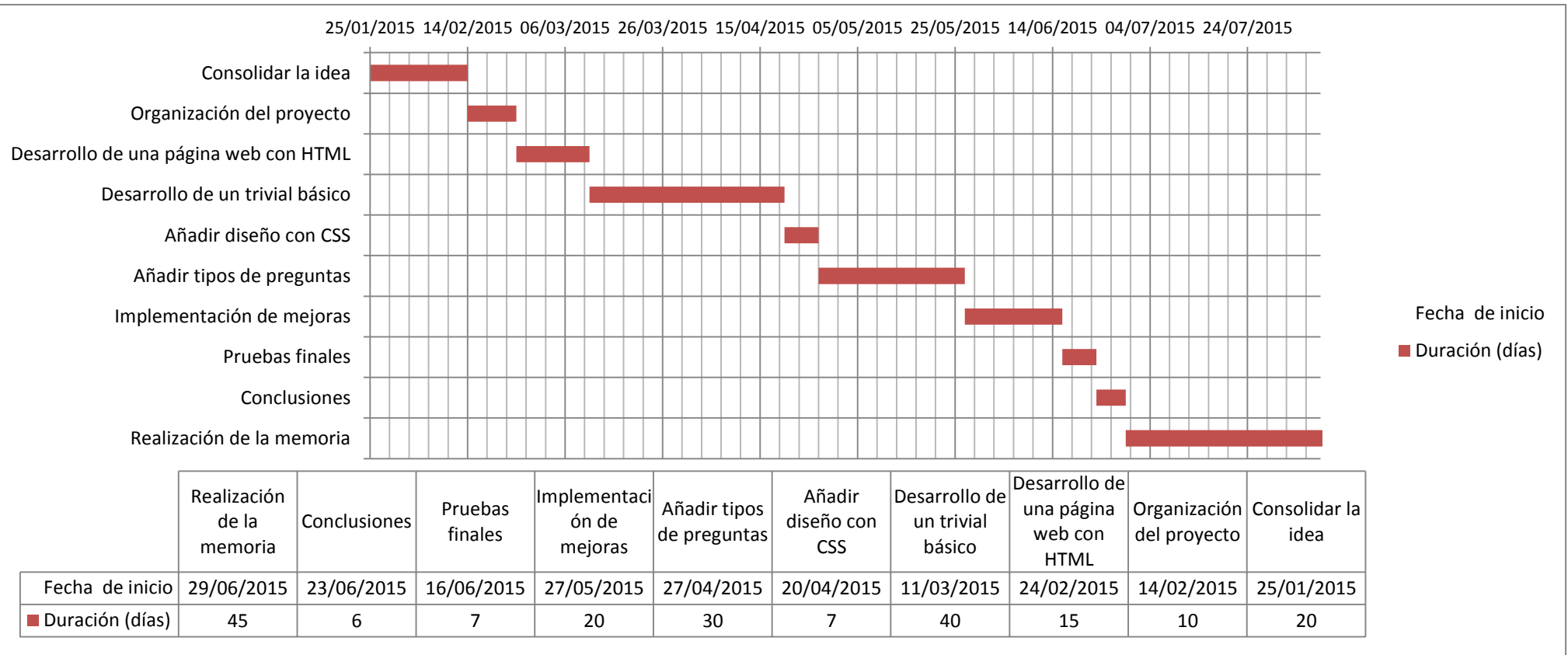


Diagrama de Gantt final

## 5.3 Comparativa del trabajo estimado y realizado

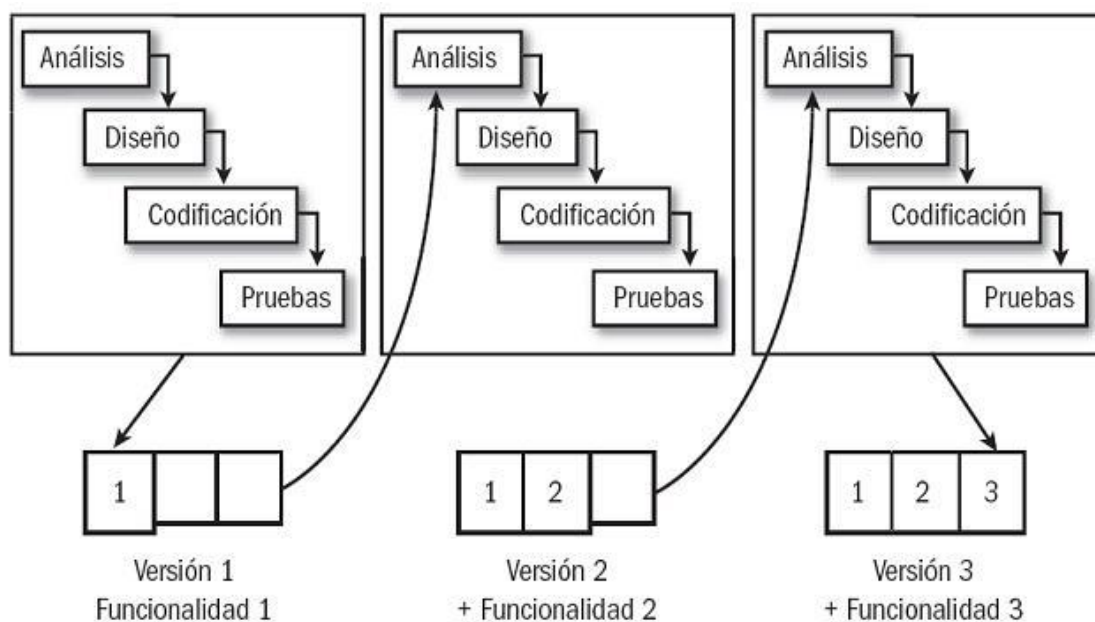
Como se puede observar, la diferencia fundamental entre ambas planificaciones radica principalmente en las actividades de consolidación de la idea, desarrollo de un trivial básico, añadir tipos de preguntas y realización de la memoria. Parte de esta ampliación de fechas ha sido debida a lo anteriormente ya comentado, problemas de trabajo y prácticas.

No existe ninguna diferencia significativa entre ambas planificaciones más allá de la gran diferencia de tiempo empleado en su realización.

Planificación	Duración (días)	Fecha inicio	Fecha fin
Inicial	141	25/01/2015	15/06/2015
Final	200	25/01/2015	13/08/2015
Porcentaje de variación	<b>4.18%</b>		

## 5.4 Ciclo de vida del proyecto

Para la realización de este proyecto se ha seguido **un ciclo de vida incremental**, puesto que el proyecto ha sido desarrollado en pequeñas partes utilizables. Se comenzó así a desarrollar un trivial básico y sencillo para después ir introduciendo pequeñas modificaciones y por tanto, una mayor complejidad. Centrándome así en la entrega de un producto operacional con cada incremento, para controlar mejor la falta de tiempo.



El diagrama previo muestra de una manera bastante aproximada el método de desarrollo seguido. Tras terminar una funcionalidad independiente, se fueron añadiendo las siguientes.

En la fase de **análisis** se establecieron las restricciones de prestaciones, planificación temporal y costes para lograr dar consistencia al proyecto y evitar así sorpresas. En esta fase, se fijaron los requisitos funcionales del proyecto, se realizó un estudio previo del proyecto y una formación en las distintas herramientas utilizadas.

En la fase de **diseño** se empezó a desarrollar una solución que cumpliera un mínimo de requisitos para poder tener ya una funcionalidad independiente. Además se elaboró una estrategia de pruebas que permitiera detectar, en una fase posterior, el incumplimiento de los requisitos por parte de la solución adoptada para así proceder a su corrección. También se gestionó dicha fase de acuerdo al plan de proyecto dentro del coste y plazo asignado.



En la fase de **codificación**, el esfuerzo se centró en desarrollar el proyecto de acuerdo al diseño de la fase anterior, de manera que éste no perdiera sus características.

En la fase de **pruebas**, se elaboró un plan de pruebas de acuerdo a la estrategia definida en la fase de diseño. Para ello, se procedió a los siguientes puntos:

- Revisar la estrategia de acuerdo al diseño realizado definiendo los diferentes niveles de prueba.
- Elaborar procedimientos de prueba para los diferentes niveles.
- Identificar los equipos y útiles de ensayo precisos.

En esta fase además, se validó y depuró el diseño y en los casos en los que fue necesario, se procedió a su modificación.

Información obtenida de [19], [20] y [21].

## 6 Análisis económico

A continuación se detallarán los costes relacionados con el desarrollo del proyecto. Para ello, se hará uso de la planificación planteada en el apartado 5.2, más todos los gastos derivados de los elementos necesarios para su elaboración. Únicamente se mostrarán los costes estimados y no los reales.

### 6.1 Herramientas hardware

En la tabla de abajo se detallan todos los medios empleados para el desarrollo de la aplicación y el coste de cada uno de ellos.

Como herramientas de tipo hardware se han empleado los siguientes dispositivos:

Dispositivo	Tarea desempeñada	Coste (€)
<b>Ordenador MacBook Pro</b>	Desarrollo de la codificación del proyecto y pruebas.	1100
<b>Ordenador Toshiba Satellite Pro</b>	Realización de la memoria	500
<b>Tablet iPad Mini 3</b>	Pruebas	400
<b>Teléfono iPhone 4S</b>	Pruebas	200
<b>Teléfono Samsung S5</b>	Pruebas	370

Cabe puntualizar que debido a que el desarrollo del proyecto ha sido llevado a cabo por una sola persona, habría sido suficiente la utilización de un único ordenador portátil, así como la de un único teléfono móvil.

Además, a la suma de los costes de todos los dispositivos habría que restar su amortización. Teniendo en cuenta que en unos 4 años trabajando se habrían amortizado todos los equipos, la amortización del coste total de todos los dispositivos (2.570 €) después de un año habría sido de:

$$2.570 - 642,5 = 1.927,5 \text{ €}$$

Por lo que si tenemos en cuenta la planificación inicial, explicada en el apartado 5.2, en la cual la duración del proyecto es aproximadamente de unos 6 meses, después de dicho periodo de tiempo quedaría un valor residual de:

$$2.570 - 321,25 = \mathbf{2.248,75 \text{ €}}$$

## 6.2 Herramientas software

A continuación se mostrarán las diferentes herramientas de tipo software empleadas para llevar a cabo el desarrollo de la aplicación.

Herramienta	Tipo de software	Coste (€)
<b>OS X 10.9.5</b>	Sistema operativo	-
<b>Windows 8</b>	Sistema operativo	120
<b>Bluefish</b>	Edición de código HTML, CSS, JavaScript y PHP	-
<b>phpMyAdmin</b>	Base de datos	-
<b>Adobe Photoshop CS6</b>	Editor de imágenes	940
<b>Microsoft Word 2010</b>	Procesador de texto	80
<b>Microsoft Excel 2010</b>	Hoja de cálculo	80
<b>Servidor de pruebas</b>	Servidor donde probar el proyecto	-

## 6.3 Gastos indirectos

Estos gastos hacen referencia a costes operacionales como por ejemplo el material de oficina o la impresión de los elementos necesarios. En este apartado se incluyen también los gastos de electricidad, alquiler y amortización inmobiliaria. Para no realizar una lista demasiado detallada de cada uno de los elementos, se añadirá como un 6% sobre el gasto del personal durante el tiempo de desarrollo del proyecto.

## 6.4 Gastos de personal

Todo el proyecto ha sido realizado por un único ingeniero del grado de ingeniería de sistemas audiovisuales adoptando diferentes roles en el proyecto. Al no haber una clara diferencia en algunos de los apartados para saber cómo imputar las horas, se ha considerado realizar una media estimada al esfuerzo necesitado. Por lo tanto, el coste por hora variará según la siguiente tabla:

Cargo	Porcentaje aproximad	Coste(€) / Hora aplicado al tanto por ciento
Tester	12%	0.54
Programador	50%	3.2
Analista	20%	1.8
Jefe de proyecto	18%	2.25

De esta forma, queda un coste total de 7.79 euros/hora.

Calculando que al día se ha trabajado en el proyecto unas 2 horas, sin contar los fines de semana y partiendo de los datos de la planificación inicial del apartado 5.2, las horas empleadas en el proyecto son:

180 días x 2 horas/día = 360 horas

Y el coste total, por lo tanto será:

360 horas x 7.79 euro/hora = **2.804,4 €**

## 6.5 Resumen de presupuesto

Concepto / Recurso	Total (€)
Recursos hardware	2.248,75
Recursos software	1.220
Gastos de personal	2.804,4
Gastos indirectos (6% de gastos totales de personal)	673

Teniendo en cuenta los cálculos de la tabla anterior, tendremos un presupuesto de unos 6.945,75 €.

A esta cifra habría que añadir un margen para imprevistos, igual a un 10% del coste anterior, y unos beneficios, igual a un 15% también del coste anterior, para poder calcular el precio neto.

Margen de imprevistos = 694,75 €

Beneficios = 1.041,86 €

Su precio neto será, por tanto:

Precio neto = 6.945,75 + 694,75 + 1.041,86 = 8.682,36 €

Ahora lo último que nos quedaría es añadir el I.V.A. (Impuesto al Valor Agregado) para conseguir el presupuesto total:

I.V.A. (21%) = 1.823,29 €

**Quedando un presupuesto total con I.V.A. igual a 10.505,65 €**

En estos gastos, no se ha tenido en cuenta los 2 años que por ley, habría que llevar el mantenimiento de la aplicación.

## 7 Análisis de la forma de venta de la aplicación

Para que una aplicación tenga éxito, lo primero que se debe hacer es pensar un modelo de negocio. Mucha gente confía tanto en el éxito de su aplicación que piensa que ya tendrán tiempo de ver como la monetizan, o que todo será tan sencillo y rápido como vendérsela a Google o Facebook.

Para que todo tenga una coherencia y un sentido, lo correcto sería que al idear y conceptualizar una aplicación, veamos desde el primer momento la vía de monetización y que construyamos el proyecto en base a ésta.

Una de las preguntas más frecuentes es qué modelo de negocio aplicar a una aplicación para conseguir una buena monetización. Hay diferentes maneras conseguirlo, algunas de ellas se describen a continuación.

### 7.1 Modelo de pago

Consiste en subir la aplicación a una tienda y asignarla un precio, sin embargo, con la cantidad de aplicaciones que hay, esta opción no es muy recomendable. Para muchos usuarios, pagar por descargarse una aplicación supone una fuerte barrera de entrada y hay que conseguir un gran número de descargas para que sea rentable.

Por estas razones, poco a poco esta opción ha ido perdiendo peso en el mercado y se ha convertido en un modelo residual, únicamente rentable para un tipo de aplicaciones muy concretas.

Para saber si una aplicación es apta para adoptar este modelo, es conveniente responder a las siguientes preguntas:

- ¿Es una aplicación de interés general o va dirigida a un público muy concreto?
- ¿Tu aplicación tiene interés a nivel mundial?
- ¿Existen más aplicaciones que hagan lo mismo que la tuya o se puede plagiar fácilmente?
- ¿Necesitas que la app sea descargada y usada por mucha gente para que sea útil?

El mercado de las aplicaciones móviles está madurando a pasos agigantados y actualmente el pago por descarga se está viendo desbancado por las “in-app purchase” (compras dentro de la aplicación).

## **7.2 Modelo de suscripción**

La aplicación sería de descarga gratuita, pero se tendría que pagar una suscripción temporal por los servicios, logrando así ingresos recurrentes.

## **7.3 Modelo por publicidad**

En esta opción el dinero se ganará por los banners de publicidad integrados en la propia aplicación. Lo más conveniente sería hacer que la aplicación fuese gratuita y posteriormente poner algunos de sus servicios de pago.

## **7.4 Modelo de compras dentro de la aplicación**

Este modelo consiste en ofrecer una aplicación de forma gratuita consiguiendo así una mayor difusión. La monetización se realiza dentro de la app mediante pequeños pagos a cambio de funcionalidades extra. De esta forma es posible conseguir que un usuario acabe pagando más por la aplicación de lo que habría gastado haciendo un único pago inicial al descargarse la aplicación.

En este modelo hay una serie de aspectos a tener en cuenta:

No debes vender tu aplicación como gratuita si obligas al usuario a realizar un pago “in-app purchase” para hacer un uso básico de la aplicación. Esto está muy mal visto y serás duramente reprendido en los reviews de los usuarios. La estrategia debe ser ofrecer una aplicación funcional desde el principio reservando las funcionalidades extra para los que quieran pagar por ellas.

Hay que cuidar y mantener al usuario. La monetización corre a cargo de los verdaderos usuarios de tu aplicación, por lo tanto es más bien una estrategia a medio/largo plazo. En el pago por descarga no importa si el usuario nunca abre la aplicación, su decisión está ligada a los reviews de otros usuarios y a la imagen que transmitas a través del icono, nombre y descripción de la aplicación. Sin embargo con un modelo basado en

“in-app-purchases” hay que ganarse al usuario día a día, es la mejor forma para hacer que tus beneficios crezcan exponencialmente.

Estudia bien los precios. Si tienes varios productos con diferentes precios te interesa que las compras se repartan equitativamente y que no todas correspondan al producto más barato ni al más caro, ya que en ese caso estarías canibalizando el resto de productos y no maximizarás los ingresos.

Plantéate, replantéate y re-replantéate si los usuarios están dispuestos a pagar por lo que les ofreces como funcionalidad extra.

## 7.5 Usuarios a los que va dirigida

La elección del modelo de negocio va unida a la base de usuario que necesitas para hacer rentable un proyecto.

El primer paso es ser totalmente consciente de las reglas que rigen el esquema de monetización que has elegido, y ver si las proyecciones de número de usuarios que manejas responden a estos esquemas.

Por ejemplo, si te decantas por un modelo basado en in-app purchases tienes que partir de la base de que sólo el 5% de los usuarios va a realizar un pago en tu app, y que la distribución de este 5% muy posiblemente se corresponda con un 3% de usuarios que realizan un único pago, un 1,5% que gaste de forma esporádica y un 0,5% cuyo gasto sea considerablemente grande.

Si no aspiras a tener una base de usuarios lo suficientemente grande como para mantener tu negocio sólo con el 5% de los usuarios deberás plantearte otros esquemas de monetización.

También es importante mejorar el ratio de retención para obtener un gran flujo de usuarios y lograr así unos buenos resultados económicos.

Puesto que sin una buena retención es prácticamente imposible hacer negocio con una aplicación. Por eso es muy importante trabajar en distintos mecanismos dentro y fuera de la aplicación que repercutan en la retención.

Conseguir una gran base de usuarios es fácil, pero muy caro. Por eso es importante que uses todos los medios posibles para adquirir usuarios, sobre todo si no dispones de muchos recursos económicos.



Algunas de las estrategias de adquisición que puedes implementar son: usar promociones cruzadas si tienes otras apps que compartan cierta afinidad o lo más importante, mejorar el posicionamiento en los markets (ASO).

## 7.6 Calcular el LTV (Lifetime Customer Value)

Una vez lanzada la aplicación, ya dispones de una serie de herramientas que te dan una idea de por dónde va.

Sin lugar a dudas, el indicativo más importante es el LTV (Lifetime Customer Value), el cual es una relación entre la retención y los ingresos por usuario.

Gracias a este indicativo tendrás una proyección de cuánto vas a ingresar por usuario a lo largo de su ciclo de vida. Con este dato podrás saber que esperar de tu aplicación, cuantos usuarios necesitas y cuanto te puedes gastar en el proceso de adquisición por cada uno de ellos.

A parte, debes integrar un sistema de analíticas en condiciones, y para que sea efectivo tendrás que detectar, definir y medir los flujos de interacción más decisivos en la app.

Así en base a estas analíticas podrás detectar cosas tan fundamentales como por qué los usuarios no finalizan un proceso de compra, no pasan de un formulario de registro o, en algunos casos, simplemente por qué dejan de usar tu app.

Una vez que has integrado los sistemas de analítica oportunos y has calculado el LTV estás en disposición de saber cuánto puedes gastarte en campañas CPI (Coste por Instalación). Si tu modelo se basa en la escalabilidad tienes que ser consciente que esto te costará dinero.

La ventaja es que gracias al LTV sabrás exactamente cuánto puedes gastar por cada usuario para que tu negocio crezca y sea sostenible.

Información obtenida de [22] y [23].

## 8 Evaluación de la aplicación

Una vez explicado el diseño y la implementación, sólo nos queda el comprobar que la aplicación cumple los requisitos establecidos y que funciona correctamente. Para dicha verificación se han realizado dos tipos de pruebas que detallaré a continuación.

### 8.1 Método de evaluación mediante cuestionario

Para evaluar el producto, se creó un formulario, accesible desde la aplicación, con preguntas relacionadas con el nivel de satisfacción que los usuarios tenían respecto a la aplicación. Sin embargo, por falta de tiempo, dicha encuesta no fue realizada por usuarios reales.

Preguntas	Puntuación
¿Le ha parecido divertida la aplicación?	
¿Volvería a utilizarla?	
¿Se la recomendaría a un amigo?	
¿Le ha parecido útil?	
¿Ha tenido alguna incidencia a la hora de jugar?	
¿Ha tenido algún problema a la hora de cambiar de idioma?	
¿Las instrucciones del juego están lo suficientemente claras y bien explicadas?	

También se ha pensado en incluir un campo en el que los usuarios puedan enviar sugerencias de la aplicación o errores encontrados en alguna de las preguntas del juego.

## 8.2 Método de evaluación mediante pruebas

Se han realizado una serie de pruebas para cerciorarnos de que la aplicación funciona correctamente, dichas pruebas son las que aparecen a continuación:

Prueba	Funcionalidad	Resultado
Al introducir la dirección web correspondiente a la aplicación, se puede acceder sin problemas.	Acceso	OK
En la pantalla principal aparecen los botones de: <ul style="list-style-type: none"><li>• “Jugar”</li><li>• “Instrucciones”</li><li>• “Información”</li><li>• “Puntuaciones”</li><li>• “Encuesta”</li><li>• “Configuración”</li></ul>	Pantalla principal	OK
Al pulsar el botón “Instrucciones”, aparece una pantalla en la que se explican todos los tipos de preguntas/respuestas de la aplicación	Instrucciones	OK
Al pulsar el botón “Información”, aparece un texto explicando el objetivo de la aplicación.	Información	OK
Al pulsar el botón “Puntuaciones”, aparece una lista con las puntuaciones de todos los usuarios ordenadas de forma decreciente.	Puntuaciones	OK
Al terminar una partida y escribir un nombre de usuario, la puntuación se mostrará en esta pantalla junto con el nombre correspondiente.	Puntuaciones	OK
Al pulsar el botón “Encuesta” aparecen una serie de preguntas a las que el usuario puede puntuar del 1 al 10.	Encuesta	OK
La encuesta se envía y se recibe correctamente.	Encuesta	OK
Al pulsar el botón “Configuración”, tienes la posibilidad de: <ul style="list-style-type: none"><li>• “Elegir un idioma” con las opciones Español/Inglés</li><li>• “Elegir dificultad” con las opciones Alta/Normal/Baja</li></ul>	Configuración	OK
Al pulsar el botón “Elegir un idioma” y elegir uno u otro idioma, se modifica también el idioma de la aplicación.	Elegir un idioma	OK
Al pulsar el botón “Elegir dificultad” y elegir una de las dificultades, las preguntas mostradas en el juego serán de dicha dificultad.	Elegir dificultad	OK
Al pulsar el botón de “Volver” en las pantallas de “Instrucciones”, “Puntuaciones”, “Encuesta” o “Configuración”, te llevará a la pantalla principal.	Botón volver	OK

Al pulsar el botón “Jugar” te lleva a una pantalla con los siguientes botones: <ul style="list-style-type: none"> <li>• “Matemáticas”</li> <li>• “Historia”</li> <li>• “Lengua”</li> <li>• “Inglés”</li> </ul>	Elegir un tema	OK
Al pulsar uno de los 4 botones anteriores, de la pantalla de “Elegir un tema”, únicamente aparecerán preguntas de la categoría seleccionada.	Elegir un tema	OK
Preguntas de tipo 1 -> estas preguntas están compuestas por: <ul style="list-style-type: none"> <li>• Pregunta: consta del texto de la pregunta y puede a parte contener o no una imagen.</li> <li>• Respuesta: consta de 4 botones de texto con cada una de las respuestas.</li> </ul>	Preguntas tipo 1	OK
Preguntas de tipo 2 -> estas preguntas están compuestas por: <ul style="list-style-type: none"> <li>• Pregunta: consta del texto de la pregunta y puede a parte contener o no una imagen.</li> <li>• Respuesta: consta de 4 imágenes correspondientes a cada una de las posibles respuestas.</li> </ul>	Preguntas tipo 2	OK
Preguntas de tipo 3 -> estas preguntas están compuestas por: <ul style="list-style-type: none"> <li>• Pregunta: consta del texto de la pregunta y un video que servirá de ayuda para responder correctamente.</li> <li>• Respuesta: consta de 4 botones de texto con cada una de las respuestas.</li> </ul>	Preguntas tipo 3	OK
Preguntas de tipo 4 -> estas preguntas están compuestas por: <ul style="list-style-type: none"> <li>• Texto explicativo de la imagen a completar.</li> <li>• Imagen dividida en 9 piezas descolocadas.</li> <li>• Cuadrado en el cual se deben poner las piezas que compone la imagen.</li> </ul>	Preguntas tipo 4	OK
Al responder cada una de las preguntas, aparece un mensaje informándote de si has respondido o no correctamente y en este último caso, te comunicará la respuesta correcta.	Mensaje informativo	OK
Al finalizar una partida, te mostrará un mensaje informativo con la puntuación total obtenida.	Mensaje informativo	OK
Después del mensaje informativo anterior,	Puntuaciones	OK

aparecerá una pantalla para introducir el nombre del usuario y guardarlo así junto a su puntuación obtenida.

## 9 Conclusiones

### 9.1 Problemas encontrados

A lo largo del proyecto, se han ido encontrando diversos problemas, a continuación se detallarán algunos de los más importantes:

- Surgieron bastantes dificultades al empezar a diseñar la base del trivial puesto que no tenía claros algunos conceptos de las tecnologías utilizadas.
- Cuando se llegaba a una pregunta que contenía una imagen y se pasaba a la siguiente, la imagen no se borraba y permanecía en la pantalla. Por lo que si la pregunta siguiente tenía también una imagen, en la pantalla aparecían la imagen de dicha pregunta y la anterior, una debajo de otra.
- Se han tenido varios problemas de incompatibilidades de bibliotecas, sobre todo a la hora de incorporar jQuery UI al proyecto.
- Hubo problemas al querer pasar un parámetro de un HTML a otro al no conocer la existencia de los parámetros jQuery.
- Al empezar a desarrollar el tipo 2 de preguntas/respuestas en el cual las respuestas ya no están compuestas por texto, sino por imágenes, surgieron algunos problemas. Al principio únicamente conseguía mostrar una de las 4 imágenes porque se superponían.
- Al desarrollar el tipo 4 de preguntas/respuestas cuyo objetivo es construir un puzzle surgieron bastantes problemas a la hora de probarlo en dispositivos móviles debido a que los eventos de arrastrar y soltar eran distintos.
- Problemas con la colocación de los elementos de la aplicación con CSS sin que la aplicación pierda su propiedad responsive.
- Al utilizar un dispositivo móvil, cuando un usuario está en una de las preguntas con el tiempo corriendo, al hacer scroll hacia abajo con el dedo y mantenerlo en la pantalla, la barra de tiempo se para. Este error no se ha conseguido solucionar a día de hoy.
- Al finalizar el juego, cuando te pide introducir un nombre para guardar la puntuación obtenida, al pulsar la tecla "Intro" el dato no se guardaba y además la página se recargaba.

Prácticamente todos los errores han sido solucionados a lo largo del desarrollo.

## 9.2 Líneas futuras de trabajo

Hay una serie de mejoras que debido a la falta de tiempo no me ha sido posible implementar, algunas de ellas habrían sido las siguientes:

- La principal sería añadir muchas más preguntas de cada tema y tipo.
- Me habría gustado crear un chat para que los distintos usuarios se comuniquen entre sí.
- Crear duelos entre jugadores.
- Mostrar gráficas con el tanto por ciento de preguntas acertadas y erróneas de cada pregunta.
- Tener una interfaz del tablero del trivial en la que poder mover las fichas y poner “quesitos” cuando un jugador los gane. Esta mejora no se ha llevado a cabo debido a la falta de conocimientos de diseño gráfico.
- Se ha dejado medio implementada la opción de poder elegir la dificultad de las preguntas en la pantalla de configuración.
- Que el usuario pueda elegir el tiempo que tiene a la hora de contestar las preguntas del juego.
- Implementar un sistema de encuestas por comunidad autónoma para poder ver los resultados obtenidos en cada una de ellas y compararlos. Para esta mejora habría sido necesario el uso de la aplicación por bastantes usuarios de las distintas comunidades.
- Dotar de robustez al código, puesto que actualmente es vulnerable a ataques informáticos como “Inyección SQL”, que podría permitir a un atacante alterar la base de datos desde fuera, hasta el punto de borrarla completamente, o “Cross Site Scripting”, introducción de scripts en la base de datos que pueden afectar al comportamiento de la web.
- Desarrollar un nuevo tipo de pregunta/respuestas llamado “Curiosidades” con la finalidad de hacer ciertos aspectos de la física más interesante para los niños. Un ejemplo de este tipo de preguntas habría sido la siguiente imagen:

## 10. Alternativas de diseño

Una vez terminada la aplicación y en estado operativo, puedo afirmar que he adquirido conocimientos de las bases sobre las que se establece la tecnología HTML5. A continuación veremos qué tipos de aplicaciones móviles hay así como las ventajas e inconvenientes de cada una de las opciones.



### 10.1 Aplicaciones nativas

Una aplicación nativa es aquella que se desarrolla de forma específica para un determinado sistema operativo, llamado Software Development Kit o SDK. Cada una de las plataformas Android, iOS o Windows Phone, tienen su propio sistema. Por lo tanto, si quieres que tu aplicación esté disponible en todas las plataformas, deberás crear varias aplicaciones con el lenguaje del sistema operativo de cada una de ellas. Los lenguajes de las plataformas más conocidas son:

- Objective-C para aplicaciones iOS.
- Java para aplicaciones Android.
- .Net para aplicaciones Windows Phone.



La principal ventaja de las aplicaciones nativas respecto a los otros dos tipos es que tienen la posibilidad de acceder a las características del hardware móvil como por ejemplo la cámara, el GPS o la agenda. Consiguiendo de esta manera que la experiencia del usuario sea mucho más positiva que con los otros tipos. Además las aplicaciones nativas no necesitan conexión a internet para poder utilizarlas.

La descarga y posterior instalación de estas app se hace siempre a través de las tiendas de aplicaciones, lo que facilita el proceso de marketing y promoción, muy importante para dar visibilidad a una aplicación.

Por lo tanto, si el coste no es un obstáculo, la mejor opción para una empresa será el desarrollo de una aplicación nativa. Pero si por el contrario, tienes un presupuesto limitado, las aplicaciones web tienen también sus ventajas.

Las principales ventajas y desventajas de este tipo de aplicaciones las podemos ver en la siguiente tabla:

Ventajas	Inconvenientes
Acceso completo al dispositivo	Diferentes habilidades/idiomas/herramientas para cada plataforma de destino
Mejor experiencia del usuario	Suelen ser más caras de desarrollar
Visibilidad en APP Store	El código no es reutilizable entre las diferentes plataformas
Envío de notificaciones o “avisos” a los usuarios	
La actualización de la app es constante	

## 10.2 Aplicaciones web

Una aplicación web, también llamada WebApp, es la que se desarrolla mediante los lenguajes HTML, JavaScript y CSS. Su principal ventaja respecto a la anterior, es la posibilidad de programar independientemente del sistema operativo en el que se usará la aplicación. Por lo que se podrá ejecutar en diferentes dispositivos sin tener que crear varias aplicaciones. Estas son las principales razones por las que se ha elegido esta opción a la hora de desarrollar este proyecto.

Las aplicaciones web se ejecutan dentro del propio navegador web del dispositivo, a través de una URL. Su mayor diferencia respecto a una aplicación nativa es que no necesita instalación y por lo tanto, no están visibles en los markets o store de los dispositivos, por lo que su comercialización se tiene que realizar de forma independiente. Aun así, cabe la posibilidad de crear un acceso directo que sería como “instalar” la aplicación en el dispositivo.

Este tipo de aplicaciones son una buena opción si nuestro objetivo es adaptar la Web a formato móvil.

Las principales ventajas y desventajas de este tipo de aplicaciones las podemos ver en la siguiente tabla:

Ventajas	Inconvenientes
El mismo código base reutilizable en múltiples plataformas, ganando además homogeneidad	Requiere de conexión a internet
Proceso de desarrollo más sencillo y económico	Acceso muy limitado a los elementos y características del hardware del dispositivo
No necesitan ninguna aprobación externa para publicarse	La experiencia del usuario (navegación, interacción...) y el tiempo de respuesta es menor que en una aplicación nativa
El usuario siempre dispone de la última versión	Requiere un mayor esfuerzo en promoción y visibilidad
Pueden reutilizarse sitios “responsive” ya diseñados	Se pierde la riqueza de personalización proporcionada desde la interfaz nativa
Suelen necesitar un menor tiempo de desarrollo que las aplicaciones nativas	

## 10.3 Aplicaciones híbridas

Estas aplicaciones son una combinación de las dos anteriores, se podría decir que recoge lo mejor de cada una de ellas. Se desarrollan con los mismos lenguajes que las WebApp, es decir, HTML, Javascript y CSS. Esto permite su uso en diferentes plataformas pero además estas aplicaciones tienen la posibilidad de acceder a gran parte de las características del hardware del dispositivo.

Su principal ventaja es que es posible agrupar los códigos y distribuir estas aplicaciones en los markets o store de los dispositivos. Una de las formas más habituales de conseguirlo es mediante phoneGap, que es un paquete de librerías que permite empaquetar aplicaciones HTML5 de manera que puedan ser usadas como aplicaciones para móviles o WebApp. En un principio, conseguir esto era el objetivo principal, pero debido a la falta de tiempo no se pudo llevar a cabo.

Las principales ventajas y desventajas de este tipo de aplicaciones las podemos ver en la siguiente tabla:

Ventajas	Inconvenientes
Es posible distribuirlas en las tiendas de iOS y Android	Experiencia del usuario más propia de la aplicación web que de la nativa
Instalación nativa pero construida con JavaScript, HTML y CSS	Diseño visual no siempre relacionado con el sistema operativo en el que se muestre

El mismo código base para múltiples plataformas

Los lenguajes web no tienen todavía compatibilidad con todas las funcionalidades/APIs nativas

Acceso a parte del hardware del dispositivo

Información obtenida de [24], [25] y [26].

## Anexo I: Manual de usuario

En el presente documento se recoge la funcionalidad completa de la aplicación “Mini trivial” disponible para web y dispositivos móviles iOS y Android. Además está desarrollado para clientes finales.

### Pantalla principal

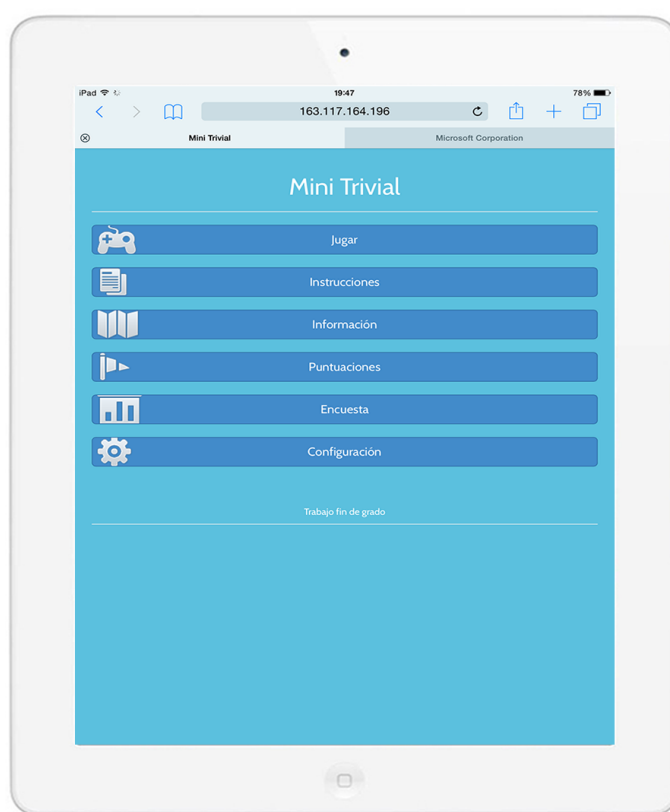


Ilustración 9: Pantalla principal

Desde esta pantalla podemos acceder a las opciones de:

- “Jugar”
- “Instrucciones”
- “Información”
- “Puntuaciones”
- “Encuesta”
- “Configuración”

## Pantalla de “Elegir un tema”

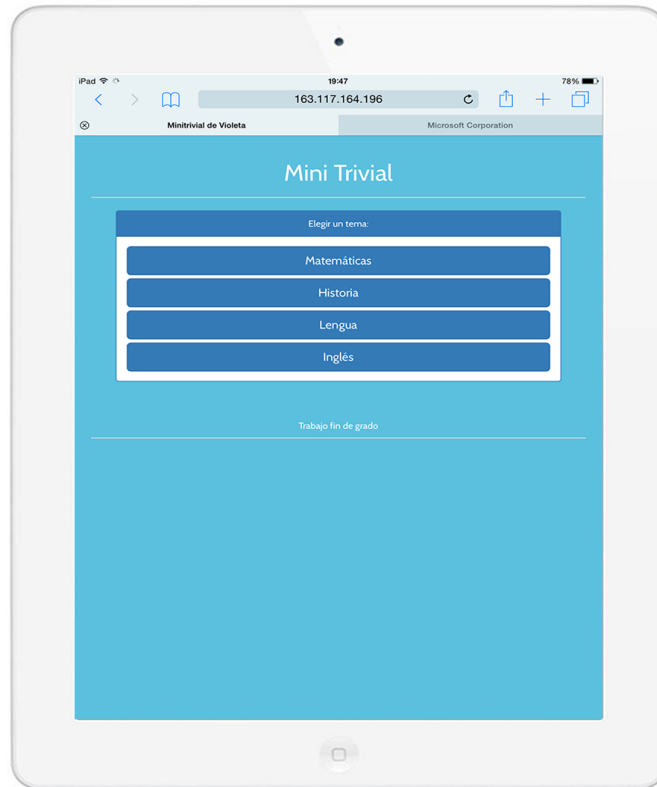
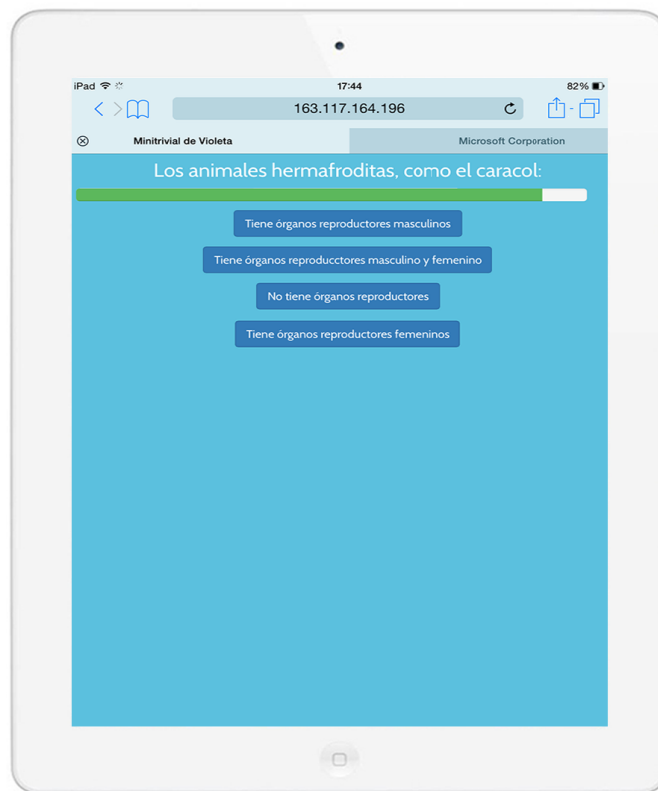


Ilustración 10: Pantalla para "Elegir un tema"

Al pulsar el botón de “Jugar”, se muestra la pantalla situada arriba en la cual puedes seleccionar uno de los siguientes temas:

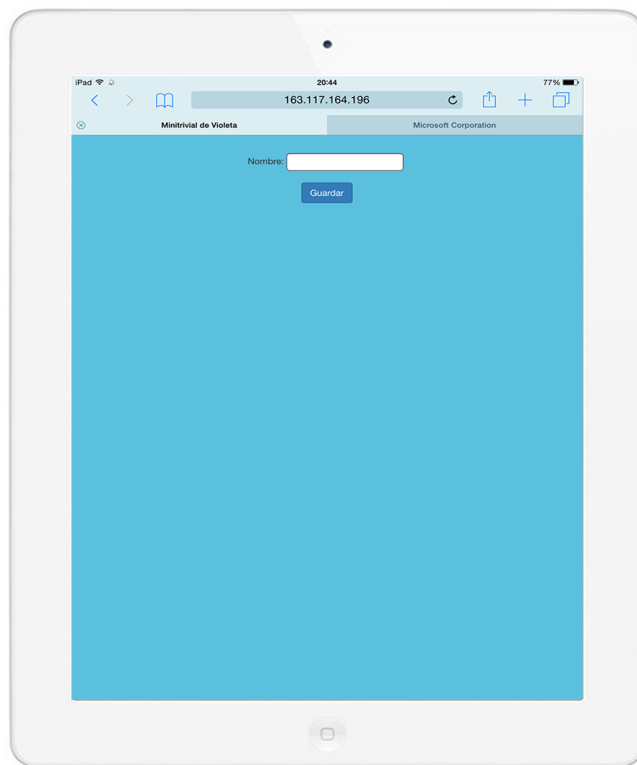
- “Matemáticas”
- “Historia”
- “Lengua”
- “Inglés”

Una vez elegida una de las opciones, empezará el juego tal y como se muestra en la pantalla situada justo debajo.



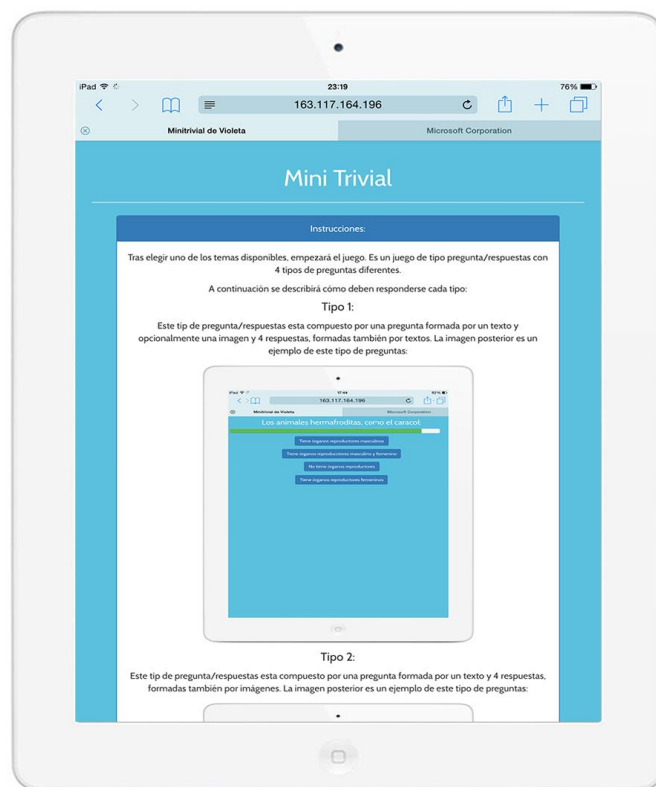
**Ilustración 11: Pantalla de juego**

Cuando ya hayas respondido a todas las preguntas, aparecerá esta pantalla en la cual podrás guardar tu nombre o el nombre que quieras introducir y que podrás consultar posteriormente en la pantalla de “Puntuaciones”.



**Ilustración 12: Pantalla de guardar puntuaciones**

## Pantalla de “Instrucciones”



**Ilustración 13: Pantalla de instrucciones**

Al pulsar el botón de “Instrucciones”, se muestra esta pantalla con texto y una serie de imágenes explicando cómo jugar y los tipos de preguntas existentes.



## Pantalla de “Información”

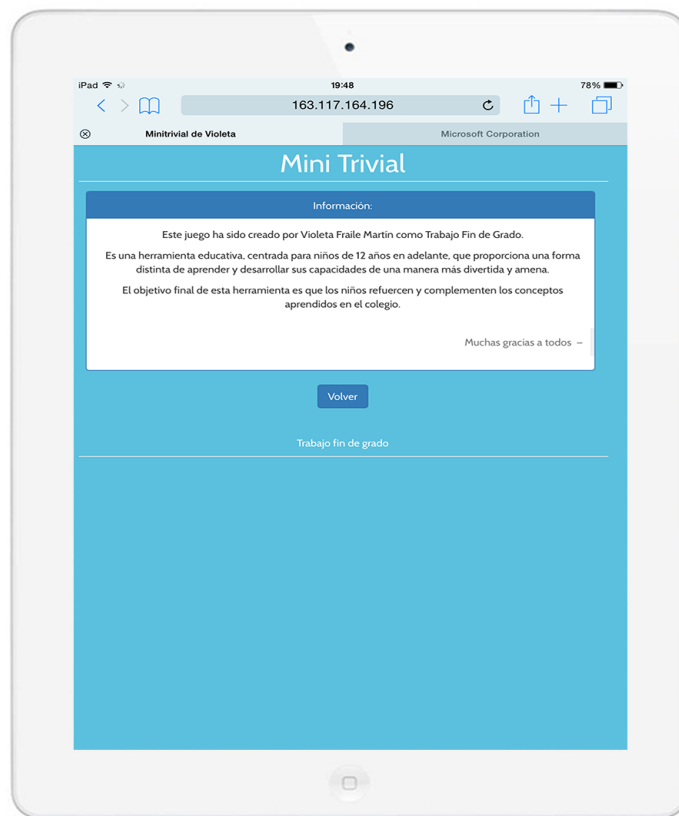
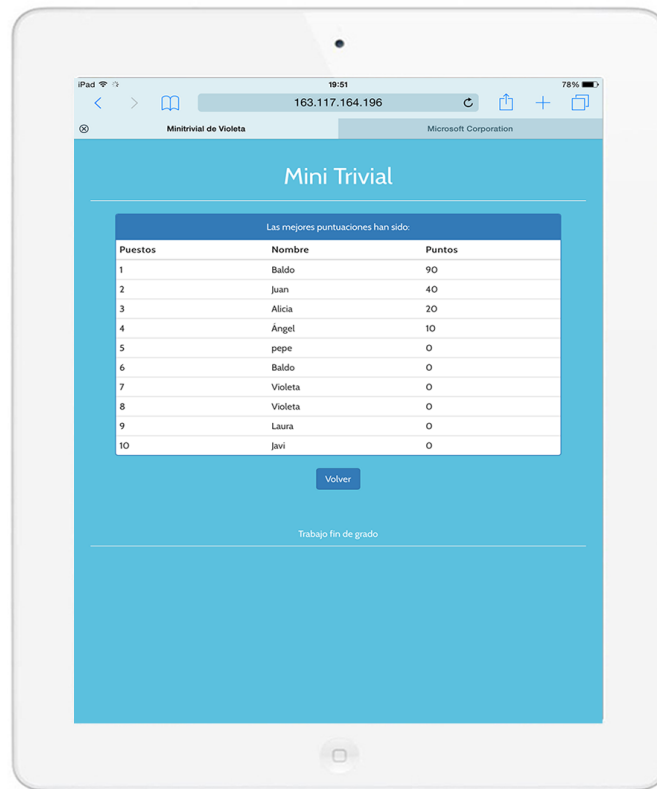


Ilustración 14: Pantalla de información

Al pulsar el botón de “Información”, se muestra esta pantalla con texto explicando las razones por las que ha sido creada la aplicación así como su objetivo.

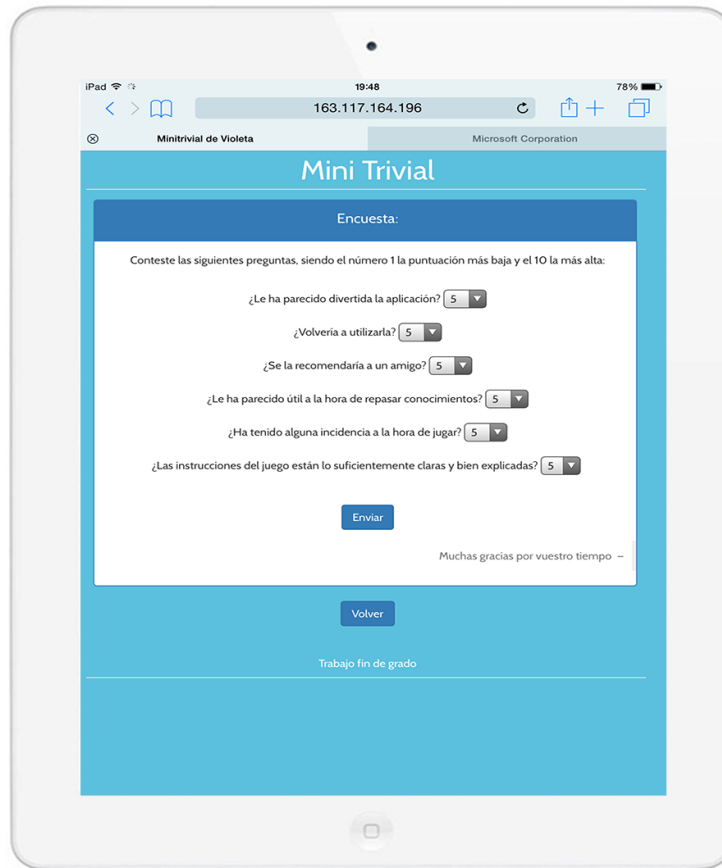
## Pantalla de “Puntuaciones”



**Ilustración 15: Pantalla de puntuaciones**

Al pulsar el botón de “Puntuaciones”, se muestran las puntuaciones de cada jugador junto con los nombres que introdujeron al guardarlas.

## Pantalla de “Encuesta”



The screenshot shows an iPad displaying a web browser with the URL 163.117.164.196. The page title is "Mini Trivial" and the header includes "Minitrivial de Violeta" and "Microsoft Corporation". The main content area is titled "Encuesta:" and contains the following text: "Conteste las siguientes preguntas, siendo el número 1 la puntuación más baja y el 10 la más alta:". Below this, there are six questions, each with a dropdown menu set to "5":

- ¿Le ha parecido divertida la aplicación?
- ¿Volvería a utilizarla?
- ¿Se la recomendaría a un amigo?
- ¿Le ha parecido útil a la hora de repasar conocimientos?
- ¿Ha tenido alguna incidencia a la hora de jugar?
- ¿Las instrucciones del juego están lo suficientemente claras y bien explicadas?

Below the questions is a blue button labeled "Enviar". Underneath the button is the text "Muchas gracias por vuestro tiempo -". At the bottom of the survey area is a blue button labeled "Volver". Below the "Volver" button is the text "Trabajo fin de grado".

Ilustración 16: Pantalla de encuesta

Al pulsar el botón de “Encuesta”, se muestra la pantalla situada arriba con preguntas para que el usuario las responda y poder así mejorar la aplicación.

## Pantalla de “Configuración”

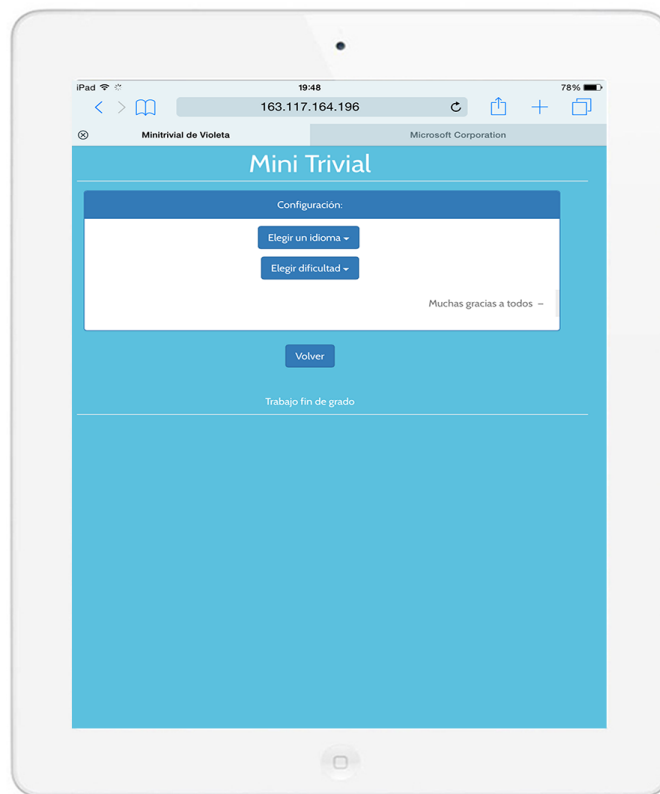


Ilustración 17: Pantalla de configuración

Al pulsar el botón de “Configuración”, se muestra esta pantalla en la que se puede configurar:

- Idioma de la aplicación: se podrá elegir entre español o inglés.
- Dificultad de las preguntas del juego: se podrá elegir entre una dificultad alta, media o baja.

## Anexo II: Summary

This project has two main objectives:

1. Learn new technologies, such as HTML5, JavaScript, CSS, JQuery, PHP and MySQL. The development of this project is an use case or a proof of concept of these languages within the field of educative videogames, and also a practical example of the possibilities they offer.
2. Make a didactic and entertaining game for all aged kids.

"Mini trivial" is a trivial-like questions and answers game, whose questions had been obtained mostly from text books from 12 year olds, so kids can learn and play at the same time.

Today, HTML has become a popular language. The advantages that it possesses had been the main reason that it was used this project. These are:

- It is an Open Source language. This is, open code and independent of third party.
- It is semantic, with tags which allow content to be classified and ordered in different levels and structures. Moreover, it also contains former metadata, which benefits the SEO, (Search Engine Optimization) positioning, and accessibility.
- Simple syntax, which ease the task for the programmer. It also allows designing lighter webs, that can be loaded quickly by the browser, giving more usability and better search engine indexing.
- It gives compatibility with mobile web browsers.
- Includes canvas paint tag, which offers more visual effects.
- Gives support to specific codecs.
- Allows video and audio embedding in a direct way.
- Makes possible user geopositioning. This is useful for mobile marketing.
- It can load web pages without being connected.
- Has new JavaScript capabilities, which raise the storing capacity. It also allows multiple JavaScript scripts running in the same page.

- Has new CSS capabilities, such as text columns, using any font on HTML, opacity, transparency, alpha channels, contrast, saturation, brightness, animation transitions, transformations, surrounded borders, gradients, shadows, etc.
- Allows to make designs for multiple electronic devices, (tablets, mobiles, computers...)

In the main screen, as we can see at the following image, we have all the possibilities that can be executed within the application. These are:

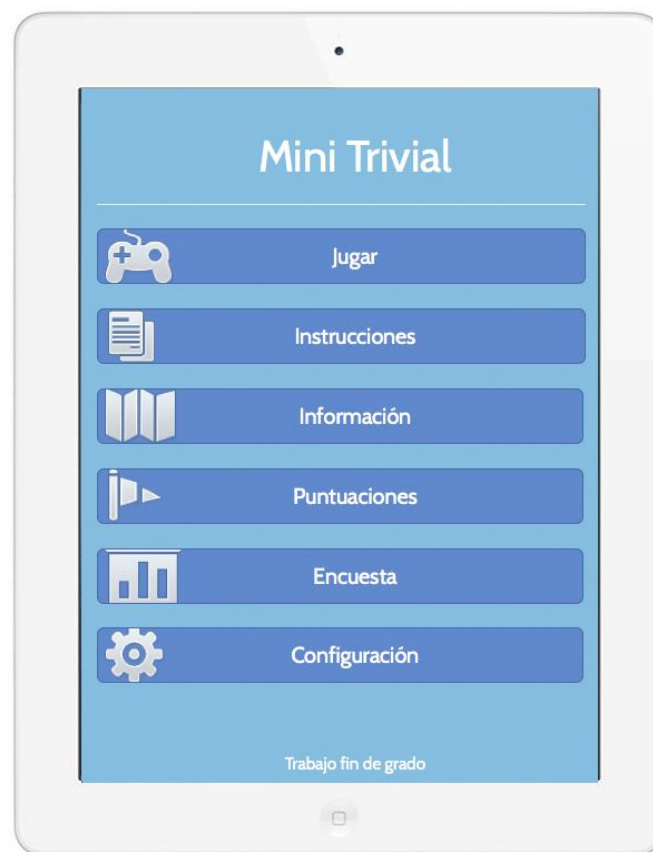


Ilustración 18: Pantalla principal

If pressing the "Play" button, we see a screen that allows the player to choose any of the four subjects corresponding with the lessons of a 12 years old kid.

Right after choosing one of them, the game will start immediately, and we can now find four kinds of questions:

1. Questions and answers written in text. In these questions the user has to choose between all of 4 possible answers, pressing his guess.

This is the first kind of question/answers implemented, and the one who took more time to code, because of the lack of skill of programming with all of the programming languages. The code was edited several times until it showed the data correctly and it progressed to the next question in the best way possible.

2. Question written and answers pictured. These kind of question makes the player choose between one of the 4 pictures shown, by clicking at the radio button located to the left side of the image.

This type of question actually took way more time than expected. Mostly caused because the great time gap between coding type 1 and type 2, it appeared some problems whenever the image was positioned. These problems appeared in computer browsers as well as in mobile devices.

3. Question and answer written, shown after a video. This kind of questions shows a short video, which contains the answer of the question that will come up right after the video comes to an end. After pressing "Continue", the next question will appear, and the user has to choose the correct answer, just like the type 1 question previously described.

This type of question was the easiest of all, thanks to the new features of HTML5. It only required a new screen that showed the video, and a transition to the question after clicking "Continue".

4. Puzzle. This question includes a torn and misplaced image, that the user has to reorder to complete the puzzle.

It was quite hard to achieve the right coding of the puzzle, but it also was easier that it looked like to make it work for computer browsers. I got the pieces to move and place in the correct boxes easily for a computer, but the mobile browsers caused severe problems. The clicking event for mobiles devices is different from the computers' so I had to change the code. Achieving that the pieces of the image moved properly was, no doubt, the hardest part of the entire coding.

Every question/answer described previously have a timer. The user has the same amount of time for the first three type of questions, but more time for the last kind of

question. This way the player, who is most likely to be a kid, has more time to complete the puzzle.

After answering a question, a message pop-up will appear, showing if the player had answered properly the question or not. In case the player gave the wrong answer, the correct answer will appear within the pop-up, so the user can learn even though he failed answering.

If the user doesn't choose any answer and therefore runs out of time, another informative pop-up will appear. This message will show an informative text to notify the player that the time came to an end, and it will also show the correct answer.

All the images that are shown by the application can be zoomed in by clicking on them, thanks to the FancyBox tool.

After pressing the "Instructions" button, located at the main page, information about the application will appear, and that will help the users to navigate through the app. Here the user can find photos and screens of the game, so they get helped in a more visual way.

If the user decides to go to "Information", also located at the main page, he will be shown a brief summary of the application info. This is, main objectives of the app, and the author that has developed it.

The game has a punctuation system as well. After finishing each round of ten questions, they get the possibility of storing their punctuation by sending their name. This is a good way to improve the competitiveness between players and to encourage them to keep playing. Also, these punctuations can be seen in the "Score" screen.

A survey section is also a part of the application. At the screen "Survey", users can rate a series of questions, which may help the software to be improved in the future.

There is a "Configuration" screen as well, including a couple of options that the user can set as its own pleasure. First of them is "Choosing a language", being Spanish and English the languages in which the application had been translated. The second option is the possibility of choosing a difficulty, which allows the user to choose between three kinds of difficulties for the game to be changed.

Finally, it is also good to note that, due to the huge grow of mobile applications, the game has been developed for a multiplatform environment, so the software works properly both mobile devices and computers.



## Anexo II: Introduction and objectives

### 1.1 HTML Introduction

Web has been changing a lot during past years. We've evolved from plain web pages, with few images, simple text and static content, to more sophisticated web pages, which now include dynamic content, graphic design, multimedia and engineering. Web pages nowadays contain online services, and they are mainly based in usefulness.

Briefly, we can define a web application as a tool in which an user, through the use of a computer, is able to request petitions to a remote application using the Internet or an intranet and receives an answer, shown by the web browser.

The meaning of HTML is HyperText Markup Language. It is a simple programming language used to design web pages, and it's easy to learn. Even though nowadays there are some other technologies to design web pages, such as VRML or Activex, HTML combined with JavaScript, are still the most used around the world. The reason being is that HTML is a well-known standard, recognized by almost every programmer. This means that all browsers would show the web design in a similar way, without inconsistencies.

An organism called World Wide Web Consortium, also known as W3C, is in charge of achieving that standardization, which means they are in charge of setting up these standards we are talking about. W3C raised the latest version of HTML in October 2014, called HTML5. In this version, new elements and attributes appeared to cover the necessities of the new Web, leaving behind HTML 4.01.

Some of the most important elements introduced in HTML5 are the following:

- `<video>` and `<audio>`: There wasn't any rule in terms of showing videos or audios in a Web page, and they could only be shown by some kind of plug-in installed. With this improvement, a new standardized format is created, so programmers now can embed videos and audios in a simpler way.
- `<article>`, `<footer>`, `<header>`, `<nav>` and `<section>`: These tags came up thanks to the use of `<div>` tags. Now, the content could be organized in a easier way, so these tags are helpful in terms of setting up the web structure.

- `<aside>`: With this tag now we can "surround" the content which isn't the main content. For example, it can be used to order the advertising banners, or some other events we want to be promoted.
- `<time>`: This tag defines a date in an established format.
- `<input>`: Improvements had been added to this tag. Now, it is possible to specify the type of data we want to get inserted in a form. We can define the text inputs as an integer, an e-mail, a date, etc.
- `<dialog>`: This tag, as suggested by its own name, defines a dialog space. Then, we can create emerging dialogs more easily.
- `<progress>`: With this tag we can represent the actual progress of an executing task. It is used along with JavaScript most of the time.
- `<canvas>`: Is a graphic container. Thanks to this tag, we can paint anything and make complex animations, with the help of JavaScript.

## 1.2 History of videogames

Videogames were born at the 40's, right after Second World War was finished. But until the 60's, modern videogames don't appear.

Nowadays there are different opinions about what was the first videogame of history. Let's check the chronology:

- In 1946 first supercomputers were built.
- In 1947 Thomas T. Goldsmith and Estley Ray Mann patented a electronic system which simulated a missile launch against a target. This project couldn't be considered as a videogame, because it didn't have any movement in the screen.
- In 1952 Alexander S. Douglas developed Nought and crosses, (OXO). This game was a computerized version of tic tac toe for the user to play against the machine.
- In 1958 William Higginbotham created Tennis for Two, a ping pong simulator. It simulated a tennis board and a net using lines and circles.
- In 19641 Rusell creates Spacewar!, a game that simulated two spaceships fighting in outer space. Players could control the spaceship direction and speed, and they were able to play against each other.

Nowadays, Spacewar! is considered as the first game of history, because Tennis for Two only used electronic circuits.

Since their beginning, videogames had evolved constantly, and they have become a solid business for industry.



Picture 19: Spacewar!

### 1.3 Objectives and motivation


I always thought that motivation is the core element of education. The best way of learning is, playing. Usually, games have some rules and some rewarding system that the users get by the time they reach the goal required by the game. These rewards guide the player to the final goal, so he can be aware of the level that he obtained, and the mistakes that he had done.

Because of this, I decided to code a trivial which questions had been obtained from books of 12 year olds. Also, the main reasons of choosing that kind of game were:

- Kids can learn lessons from their books by playing.
- Kids can check right off the bat if the answer they posted was correct or incorrect. In case of that the answer is wrong, they get the correct answer.
- Using a punctuation system, the motivation of the kids gets enhanced. In the official trivial game, that motivation comes from the plastic wedges.
- Questions apart from containing text, they have images and videos. Thanks to that, the game gets more pleasant and the kids get more enjoyed by playing the game.

By the other side, this project has two more clear objectives:

- The coding of a tool that has new features in terms of kids learning in The Internet.
- Helping kids to learn the same way they'd do with text books, but in a funnier and more dynamic way.
- Learning web programming languages, so it can help me work in the future in that kind of field.



Finally, other things to take into account to motivate players and students are developing a positive reaction with the mistakes made. In a videogame we learn about our own failure, because the mistakes are not seen as an end, but as a required step to reach a desired goal. The trivial tries to find a positive relation with the player, giving him constant feedback, and keeping individual learning chapters. In both cases, mistakes are seen as obvious; for example, answering wrongly a question shows the individual progress of each player, and it emanates self-competitiveness. This is because one of the goals of this project is showing the player the correct answer all the times, even if he fails, he learns.

Videogames are great for achieving a player getting concentrated on the game, and cheering players up to become better and better at playing it. These techniques can be also transferred to kids' classrooms, so students are more likely to pay attention.

All being said, I would like to detail that even though videogames could be a great tool for people to learn, they shouldn't become the only method of learning. That is, they are just an extra tool of learning.

## Anexo III: Conclusions

### 9.1 Problems found

During the project, some problems had been found, being the most important ones exposed:

- A lot of difficulties appeared just at the beginning. Designing the game's basis was hard, because I didn't have some concepts about the technology used clear enough.
- By the time a question with an image was reached, and the next question appeared, the image didn't get erased, and stayed in the screen. This caused even more problems if the next question was also of the same kind, because both images got messed up.
- Problems with libraries also happened. Most of them were related to JQuery UI.
- Whenever a HTML parameter wanted to be passed to another HTML, issues also appeared, because of the misunderstanding with JQuery's parameters.
- At the beginning of type 2 of questions/answers, some issues appeared as well. This happened because they were composed by images rather than text, so the code changed. The main problem happened trying to show all four images, because they rather appeared on top of each other.
- The puzzle type of questions/answers also had problems. The hardest problem to solve was the compatibility with mobile devices, because the events drag & drop were different.
- Issues with placing elements with CSS, because some other elements lost their responsive capability.
- When using a mobile device, if the user is in a time-based question and he scrolls down, the timer bar gets stopped. This bug couldn't be fixed.
- By the time the user finished the game and attempted to save his name, it wouldn't save if he pressed the enter key.

Almost all the bugs had been fixed along the development of the application.

## 9.2 Future lines of work

Unfortunately, there are some improvements to the application that couldn't be done because of the lack of time. They couldn't be coded, but can be exposed:

- Adding much more questions of each type.
- Coding duels between players.
- I was really interested in developing a chat, so users could communicate.
- Showing graphics with correct and wrong answers' percentages.
- Designing a board interface, in which users could move the game playing piece and stack wedges whenever a player answers correctly. This improvement couldn't be coded because of the lack of knowledge about graphic design.
- Setting the difficulty of the game has been semi-coded.
- Allowing the user to change the time he has to answer the questions.
- Coding a system survey for each region, so the results can be compared. Introducing this requires the application to be spread all across the country.
- Making the code securer. Right now, SQL injections can be triggered by an attacker, which could alter the databases, and even allowing him to delete the entire data. Also, Cross Site Scripting is another common attack that the application is vulnerable, allowing a potential attacker to store scripts inside the database.
- Developing a new type of questions, called "Curiosities". This kind of question would include interesting physics notions for kids. An example of this kind of questions would be the following image:

## Referencias

- [1] Ruth S. Contreras (16/04/2015), "Motivación aprendizaje y juegos". Disponible en: <http://www.startvideojuegos.com/motivacion-aprendizaje-y-juegos/>
- [2] Federico-erostarbe (30/03/2010), "HTML5: qué cambios trae la nueva versión del lenguaje estándar de la web". Disponible en: <http://hipertextual.com/archivo/2010/03/html5-que-cambios-trae-la-nueva-version-del-lenguaje-estandar-de-la-web/>
- [3] Elotrolado,"La historia de los videojuegos". Disponible en: [http://www.elotrolado.net/wiki/Historia\\_de\\_los\\_videojuegos](http://www.elotrolado.net/wiki/Historia_de_los_videojuegos)
- [4] Simone Belli (2008), "Breve historia de los videojuegos". Disponible en: [http://www.academia.edu/214747/Breve\\_historia\\_de\\_los\\_videojuegos](http://www.academia.edu/214747/Breve_historia_de_los_videojuegos)
- [5] Pablo Ruiz Soria, "Trivinet". Disponible en: <http://www.trivinet.com>
- [6] Wikipedia, "Trivial Pursuit". Disponible en: [https://es.wikipedia.org/wiki/Trivial\\_Pursuit](https://es.wikipedia.org/wiki/Trivial_Pursuit)
- [7] Testeando S.L., (2015), "Testeando". Disponible en: <http://www.testeando.es>
- [8] Universidad de Murcia, "Manual básico de creación de páginas web" <https://www.um.es/atika/documentos/html.pdf>
- [9] Bartolomé Sintés Marco (17/09/2015), "Historia de la Web: HTML y CSS". Disponible en: [http://www.mclibre.org/consultar/htmlcss/otros/otros\\_historia.html#Cascading](http://www.mclibre.org/consultar/htmlcss/otros/otros_historia.html#Cascading)
- [10] LibrosWeb (2015), "Breve historia de CSS". Disponible en: [https://librosweb.es/libro/css/capitulo\\_1/breve\\_historia\\_de\\_css.html](https://librosweb.es/libro/css/capitulo_1/breve_historia_de_css.html)

- [11] Pro Solutions Soluciones Informáticas Integrales (30/10/2013), "Ventajas y desventajas de CSS". Disponible en:  
<http://www.prosolutions.es/blog/ventajas-y-desventajas-de-css/>
- [12] LibrosWeb (2015), "Especificaciones oficiales". Disponible en:  
[http://librosweb.es/libro/javascript/capitulo\\_1/especificaciones\\_oficial.es.html](http://librosweb.es/libro/javascript/capitulo_1/especificaciones_oficial.es.html)
- [13] Mundo geek, "Una no tan breve historia de PHP". Disponible en:  
<http://mundogeek.net/archivos/2011/01/26/una-no-tan-breve-historia-de-php/>
- [14] Xavier du Tertre (06/03/2014), "¿Qué es Bootstrap? – La Historia y el Bombo: Parte 1 de 2". Disponible en:  
<https://www.prestashop.com/blog/es/que-es-bootstrap-la-historia-y-el-bombo-parte-1-de-2/>
- [15] Gonzalo Chacaltana (04/03/2014), "Una breve cronología de MySQL". Disponible en: <http://www.solocodigoweb.com/blog/2014/03/04/una-breve-cronologia-de-mysql/>
- [16] Alicia Herranz, "Cómo evitar el fracaso escolar". Disponible en:  
<http://www.serpadres.es/familia/tiempo-libre/articulo/evitar-fracaso-escolar-estudiar>
- [17] Celia Rodríguez Ruiz, "10 consejos para afrontar el fracaso escolar". Disponible en: <http://www.educapeques.com/escuela-de-padres/afrontar-fracaso-escolar.html>
- [18] Pablo García, "Cuáles son las causas del fracaso escolar". Disponible en: <http://educacion.uncomo.com/articulo/cuales-son-las-causas-del-fracaso-escolar-19512.html>
- [19] Avelino Felipe Policarpio (28/04/2012), "Modelo incremental". Disponible en: <http://es.slideshare.net/AvelinoFelipePolicarpio/modelo-incremental-12729581>



[20] Administrador del Wiki – EOI (09/05/2012), “El modelo de fases y ciclo de vida del proyecto en gestión de proyectos 2”. Disponible en:  
[http://www.eoi.es/wiki/index.php/EL\\_MODELO\\_DE\\_FASES\\_Y\\_CICLO\\_DE\\_VIDA\\_DEL\\_PROYECTO\\_en\\_Gestión\\_de\\_proyectos\\_2](http://www.eoi.es/wiki/index.php/EL_MODELO_DE_FASES_Y_CICLO_DE_VIDA_DEL_PROYECTO_en_Gestión_de_proyectos_2)

[21] EcuRed, “Ciclo de vida de un proyecto”. Disponible en:  
[http://www.ecured.cu/index.php/Ciclo\\_de\\_Vida\\_de\\_un\\_Proyecto](http://www.ecured.cu/index.php/Ciclo_de_Vida_de_un_Proyecto)

[22] Miguel Muñoz, 08/02/2012), “In app purchase: la nueva tendencia monetizando aplicaciones para móviles”. Disponible en:  
<http://www.biko2.com/movilidad/in-app-purchase-la-nueva-tendencia-monetizando-aplicaciones-para-moviles/>

[23] Miguel Muñoz (13/08/2014), “Claves de negocio para hacer que tu app sea rentable”. Disponible en: <http://www.biko2.com/movilidad/claves-de-negocio-para-hacer-que-tu-app-sea-rentable/>

[24] Jesús E. Pérez Villegas (08/02/2015), “Apps nativas vs apps basadas en web vs apps híbridas”. Disponible en:  
<http://yises.com/blog/2015/02/08/apps-nativas-vs-apps-basadas-web-vs-apps-hibridas/>

[25] Adolfo Fernández Gutierrez (20/02/2013), “Aplicaciones web vs. Aplicaciones nativas vs. Aplicaciones híbridas”. Disponible en:  
<http://blogthinkbig.com/aplicaciones-web-nativas-hibridas/>

[26] Lance Talent (20/02/xxxx), “Los 3 tipos de aplicaciones móviles: ventajas e inconvenientes”. Disponible en:  
<http://www.lancetalent.com/blog/tipos-de-aplicaciones-moviles-ventajas-inconvenientes/>

- [27] Rolandocaldas, “localStorage en HTML5. El fin de las cookies”. Disponible en: <https://rolandocaldas.com/html5/localstorage-en-html5>
- [28] Miguel Angel Alvarez (16/10/2014), “AngularJS Vs jQuery”. Disponible en: <http://www.desarrolloweb.com/articulos/angularjs-vs-jquery.html>
- [29] Wikipedia (01/06/2015), “Triviados (Atríviate)”. Disponible en: [https://es.wikipedia.org/wiki/Triviados\\_\(Atr%C3%ADviate\)](https://es.wikipedia.org/wiki/Triviados_(Atr%C3%ADviate))
- [30] Arsys (14/05/2013), “¿Qué aspectos legales debo conocer antes de lanzar una página web?”. Disponible en: <http://www.arsys.info/general/que-aspectos-legales-debo-conocer-antes-de-lanzar-una-pagina-web/>